

Technology Computer-Aided Design

Siegfried Selberherr

Institute for Microelectronics
Technical University of Vienna
Gusshausstrasse 27–29, A-1040 Vienna, Austria

Abstract

Computer simulation is an indispensable tool for the design of new VLSI devices. Besides the progress in physical models and computational techniques which has dominated *technology computer aided design* (TCAD) in the past, purely software-related aspects are attracting increasing attention and will become a crucial issue for future developments in this field.

I. Introduction

Process and device simulation is commonly used for the design of new VLSI devices and processes and as an explorative tool to gain a better understanding of process and device physics. On the other hand, simulation is also carried out after the design phase in order to optimize certain parameters of a technology, e.g., to improve device performance and reliability or to increase the yield [1].

For all these tasks the term TCAD, short for *technology computer aided design*, was born. TCAD includes both physically rigorous and simplified process and device simulation in one to three spatial dimensions. Furthermore, links to layout-oriented CAD and to circuit simulation are required.

Depending on the particular application of TCAD tools, different demands arise: for the development of new technologies and for the prediction of the behavior of new devices both accuracy and robustness are required. In this case, very sophisticated physical models and numerical techniques must be used, usually at a high computational cost. An example of such a model is the two-dimensional simulation of the transient-enhanced diffusion during rapid thermal annealing [2].

On the other hand, for statistical simulations [3] or post-design process optimizations [4], speed is the most crucial issue, as physical models can be calibrated to an existing manufacturing process and hence do not pose a reliability problem.

Independently of the progress in advanced physical modeling, the fast and simple “tuned” models will still remain in broad use; there is no unique “best model” for all simulation problems.

TCAD involves a number of scientific disciplines in addition to electrical engineering and computer science. This has also had an impact on the properties of the software which has been produced by that heterogenous community during the past 20 years.

II. Frameworks

For a long time the importance of pure software issues for TCAD has been underestimated. In the past few years, as these issues attract increasing attention, the major focus is on the integration of TCAD tools into a common framework.

II.1 TCAD versus ECAD

In the *electronic CAD* (ECAD) field, frameworks for tool integration and data interchange standards have been emerging during the past years. Hardly any satisfying framework implementations are available for TCAD, especially for advanced, multi-dimensional process and device simulation. One reason for that is the difference in the sizes of the user communities. Electronic CAD is more widely used than Technology CAD.

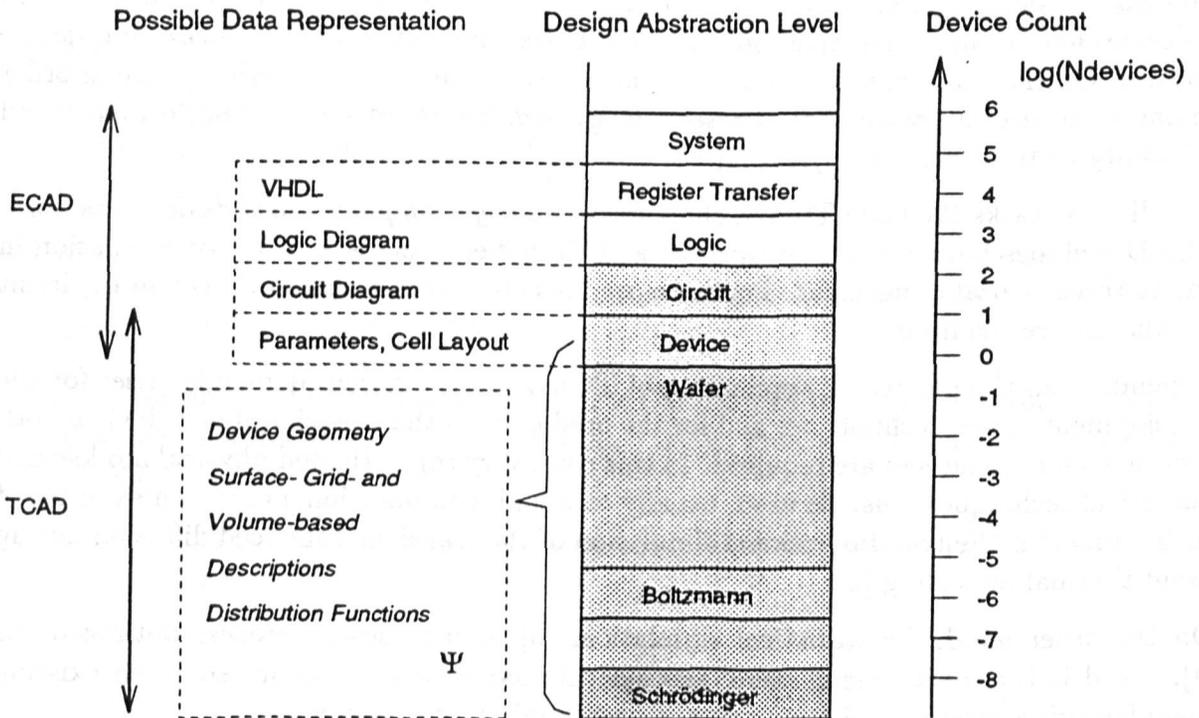


Figure 1: Design levels and device count scale in ECAD and TCAD

Another possible reason for the differences between ECAD and TCAD in terms of progress in frameworks and standardization is shown in Figure 1: in ECAD there have always been

several clearly defined layers of design abstraction. Though the device count scale is open towards the high end, there is a well-defined lower bound for ECAD, which is the single device. For every order of magnitude in device count there is a common abstraction level and at least one well-established data representation as well. For TCAD however, the only evident lower boundary in terms of abstraction is the physical atom. There are no “natural” intermediate abstraction levels and as an indirect consequence of physical modeling there is no well-established unique data representation like in the ECAD case.

These considerations indicate that finding a unified data representation for TCAD is dominated by semantic problems which are closely related to the large interval (Figure 1) of device count to be represented, together with the lack of clearly defined intermediate abstraction levels and the multi-disciplinary background involved in TCAD.

II.2 Framework Demands

As an operating environment for TCAD tools and engineers, a TCAD framework must provide the following key features:

- allow minimum effort integration of existing tools and facilitate the development of new tools
- allow casual users to use simulation in a black box manner
- provide enough flexibility on the task level to accomodate easily to new design tasks
- provide an extensible database for design representation
- be “open” in terms of platform independence, availability and the use of open standards
- provide standard functionality like visualization, interactive structure editing and postprocessing as generic tools

II.3 Existing Approaches

Facing these rather rigorous demands and the potential problems stated above, one cannot expect to find an easy and fast path leading to the ultimate TCAD framework. However, various attempts to head in that direction may be found:

In the semiconductor industry there have been emerging a number of remarkable framework efforts worldwide, such as an integrated system for statistical VLSI design [5] (Hitachi, Japan), an integrated, graphical device design environment [6] (Phillips, UK), SATURN [7] (Siemens, Germany) or the MECCA system [8] (AT&T, USA).

Based on an initial proposal by Duvall [9], there have been various *Profile Interchange Format* (PIF)-based design environments both in industry (like the PRIDE [10] system) and at universities (e.g. PROSE [11] from UC Berkely).

Commercial TCAD vendors are integrating their tools and providing them with unified user interfaces, like STUDIO from Technology Modeling Associates, or MASTERPIECE from Silvaco Data Systems.

Depending on the intentions of the creator of the frameworks, different aspects, like a rigorous task level implementation or a comfortable user interface [12], have been emphasized. Unfortunately, this has often been done at the expense of portability, or by leaving out a unified data representation.

Recently, a client-server framework architecture has been introduced by the semiconductor wafer representation working group [13] of the *CAD Framework Initiative* (CFI), an international standardization committee for ECAD. The intriguing goal of this approach is to separate the physical modeling completely from tedious tasks such as grid generation, interpolation, or geometry handling by providing these functions as a black-box server which is accessed by the simulation clients via a procedural interface. This method is very well-suited for the simulation of topography formation, however, it can be detrimental to applications with high data throughput or applications which exhibit performance advantages thanks to a tight coupling between physical models and numerical techniques.

In an attempt to address all of the framework demands stated above, we have developed VISTA, the *Viennese Integrated System for TCAD Applications*. It consists of a PIF Database, which is an enhanced intertool version of the well-known profile interchange format proposed in [9]. To accommodate the needs of existing TCAD applications, the original PIF syntax was restructured by reducing the number of different constructs, adding a few new constructs such as tensor product grid definition, and by defining additional semantic rules for the use of standardized attributes. Our PIF implementation can be used to store arbitrary LISP expressions for process flow representation.

Simulators and all other tools access the PIF database using the *PIF Application Interface* (PAI) [14], which supports several programming languages including C, FORTRAN and LISP. The PAI is a procedural interface for accessing the binary PIF database. It provides functionality for creating, reading and modifying PIF objects. In this way the application programmer does not need to know too much about the PIF syntax to be able to use the PIF database. The PAI was designed as a strictly layered product to guarantee the necessary functionality, performance and extensibility.

A *system layer* hides all system dependencies concerning communication with the operating system from the rest of the PAI. A *caching layer* takes care of performance and space requirements. The *interface layer* allows access to the PIF objects suited for advanced C and is the standardized interface to the PIF database. The application layer provides a more comfortable access to PIF objects for applications written in C, FORTRAN or LISP.

For the conversion from the binary intertool form to the intersite ASCII format the PIF binary file manager (PBFM) has been developed. In this way the intersite exchange of PIF files between different hardware and software platforms, for instance via electronic mail, is supported.

The simulators are controlled by an interpreting TCAD shell [15] which integrates all system components on the task level. The major benefits of selecting LISP as extension

language are that full-fledged programming language features like branches, loops and subprograms, mechanisms for defining new variables and standard mathematical operations and expressions can be utilized to define complex development tasks. The XLISP [16] interpreter which we have chosen as basis for the TCAD shell, is a publicly available software product written in portable C, and has the additional feature of being highly system-independent.

The user interface of VISTA [17] is based on the X Window system. By means of the X Toolkit [18] it implements most of the functionality which is required to support TCAD information flow as so-called *widgets*. The user interface is tightly coupled with the XLISP interpreter. This combination of a widget-oriented user interface with an interpreter is known to be a very flexible and promising concept [19], [20], [21].

An interactive device editor, generic postprocessing and visualization modules are also part of the framework. Special emphasis has been put on the use of open portable subsystems (which are mostly public domain products) to achieve a high degree of portability.

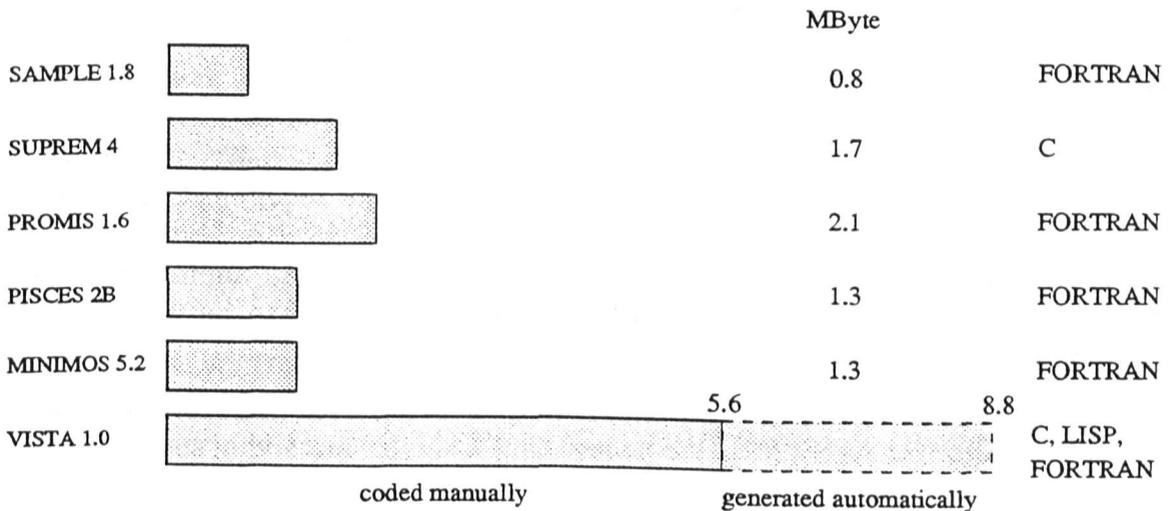


Figure 2: Comparison of source code sizes. VISTA does not include simulation tools.

Figure 2 shows the code sizes of well-known simulation tools. SAMPLE [12] (from UC Berkeley) is a topography simulation tool, MINIMOS [22] and PROMIS [2] are two-dimensional MOS device and process simulation programs, respectively, developed at TU Vienna. SUPREM4 is a two-dimensional process simulation program and PISCES is a two-dimensional device simulation program, both from Stanford University [23].

Given VISTA's remarkable code size compared to these classical single simulation tools, it becomes obvious that a framework for tool integration and development must exhibit a homogenous architecture in order to be comprehensible and maintainable.

This can be achieved through the use of unified concepts. The generalization of existing solutions within the framework should be favoured over the introduction of new ones. A second key to a comprehensible system is the use of abstract concepts. In VISTA, tedious

parts of the sourcecode, like the PAI or interface code which is required for multilanguage programming, is generated automatically from abstract specifications. Only the creative parts are left to the programmer.

III. The Future

It is a fact that the rapidly shrinking device dimensions and the increasing cost of experimental design optimization have been stimulating the evolution of TCAD; it can be expected that TCAD will continue to evolve into several directions:

The exact future development of physical models is hard to predict, as models always reflect the state of technology [24], [25]. Future models will have to be based on a lower level of abstraction. As a consequence the complexity and the “bandwidth” of models used for TCAD will increase. As to computational techniques, it is obvious that the availability of massively parallel computers will stimulate the development and use of parallelizable methods, like Monte Carlo simulation, e.g. [26].

Besides the progress in physical modeling, the future of process and device simulation will be significantly influenced by the introduction of TCAD frameworks and its impacts. This integration will allow simulation to catch up with the physical reality. It is to be hoped that TCAD frameworks will help to reduce the gap between the engineer’s simulation needs and the sophistication of the available models and simulation methods by providing a “plug-and-play” environment for tool developers. One major requirement towards this goal is the semantic standardization for TCAD data, which concerns describing process information in a comprehensive and unambiguous way rather than finding appropriate representation methods. Client-server concepts can be expected to be used in future simulation tools in different places, e.g. for solving large linear systems.

A new object-oriented method for CAD tool management has been demonstrated through the Cadwell design framework [27]. We believe that a similar method of tool abstraction should be employed in a framework for advanced process and device simulation as well.

The importance of process and device simulation will increase in general, and as a consequence of the broader use, performance, together with robustness and ease of use will become even more crucial. The integration of different process and device simulators into modern TCAD systems will become a necessity because of the inherent communication, data transfer and maintainance advantages.

Acknowledgement

The VISTA project is supported by the “Forschungsförderungsfonds für die gewerbliche Wirtschaft”, project 2/285 and 2/299, and by the research laboratories of AUSTRIAN INDUSTRIES - AMS at Unterpremstätten, Austria; DIGITAL EQUIPMENT at Hudson, USA; HITACHI at Tokyo, Japan; PHILIPS at Eindhoven, the Netherlands; SIEMENS at Munich, FRG; and SONY at Atsugi, Japan.

References

- [1] D. Cole, E. Buturla, S. Furkay, K. Varahramyan, J. Slinkman, J. Mandelman, D. Foty, O. Bula, A. Strong, J. Park, T. Linton Jr., J. Johnson, M. Fischetti, S. Laux, P. Cottrell, H. Lustig, F. Pileggi, and D. Katcoff, "The Use of Simulation in Semiconductor Technology Development," *Solid-State Electron.*, Vol. 33, No. 6, pp. 591–623, June 1990.
- [2] G. Hobler, S. Halama, W. Wimmer, S. Selberherr, and H. Pötzl, "RTA-Simulations with the 2-D Process Simulator PROMIS," in Proc. *NUPAD III Workshop*, (Honolulu), pp. 13–14, 1990.
- [3] J. Brockman and S. Director, "A Macromodeling Approach to Process Simulator Tuning," in Proc. *NUPAD III Workshop*, (Honolulu), pp. 17–18, 1990.
- [4] J. Wenstrand, H. Iwai, M. Norishima, H. Tanimoto, T. Wada, and R. Dutton, "Intelligent Simulation for Optimization of Fabrication Processes," in Proc. *NUPAD III Workshop*, (Honolulu), pp. 15–16, 1990.
- [5] H. Matsuo, H. Masuda, S. Yamamoto, and T. Toyabe, "A Supervised Process and Device Simulation for Statistical VLSI Design," in Proc. *NUPAD III Workshop*, pp. 59–60, 1990.
- [6] P. Gough, M. Johnson, P. Walker, and H. Hermans, "An Integrated Device Design Environment for Semiconductors," *IEEE Trans. Computer-Aided Design*, Vol. 10, No. 6, pp. 808–821, June 1991.
- [7] H. Jacobs, W. Hänsch, F. Hofmann, W. Jacobs, M. Paffrath, E. Rank, K. Steger, and U. Weinert, "SATURN - a Device Engineer's Tool for Optimizing MOSFET Performance and Lifetime," in Proc. *NUPAD III Workshop*, (Honolulu), pp. 55–56, 1990.
- [8] P. Lloyd, H. Dirks, E. Prendergast, and K. Singhal, "Technology CAD for Competitive Products," *IEEE Trans. Computer-Aided Design*, Vol. 9, No. 11, pp. 1209–1216, November 1990.
- [9] S. Duvall, "An Interchange Format for Process and Device Simulation," *IEEE Trans. Computer-Aided Design*, Vol. 7, No. 7, pp. 741–754, July 1988.
- [10] M. Simpson, "Pride: An Integrated Design Environment for Semiconductor Device Simulation," *IEEE Trans. Computer-Aided Design*, Vol. 10, No. 9, pp. 1163–1174, September 1991.
- [11] A. Wong, *Technology Computer-Aided Design Frameworks and the PROSE Implementation*. PhD thesis, University of California, Berkeley, 1992.
- [12] E. Scheckler, A. Wong, R. Wang, G. Chin, J. Camanga, A. Neureuther, and R. Dutton, "A Utility-Based Integrated System for Process Simulation," *IEEE Trans. Computer-Aided Design*, Vol. 11, No. 7, pp. 911–920, July 1992.

- [13] D. Boning, G. Chin, R. Cottle, W. Dietrich, S. Duvall, M. Giles, R. Harris, M. Karasick, N. Khalil, M. Law, M.J. McLennan, P. Mozumder, L. Nackman, S. Nassif, V. Rajan, D. Schröder, R. Tremain, D. Walker, R. Wang, and A. Wong, "Developing and Integrating TCAD Applications with the Semiconductor Wafer Representation," in Proc. *NUPAD IV Workshop*, (Seattle), pp. 199–204, 1992.
- [14] F. Fasching, C. Fischer, S. Selberherr, H. Stippel, W. Tuppä, and H. Read, "A PIF Implementation for TCAD Purposes," in Proc. *SISDEP 91 Conf.*, (Zürich), pp. 477–482, 1991.
- [15] H. Pimingstorfer, S. Halama, S. Selberherr, K. Wimmer, and P. Verhas, "A Technology CAD Shell," in Proc. *SISDEP 91 Conf.*, (Zürich), pp. 409–416, 1991.
- [16] D. Betz, *XLISP: An Object-Oriented LISP, Version 2.1*, 1989.
- [17] S. Halama, F. Fasching, H. Pimingstorfer, W. Tuppä, and S. Selberherr, "Consistent User Interface and Task Level Architecture of a TCAD System," in Proc. *NUPAD IV Workshop*, (Seattle), pp. 237–242, 1992.
- [18] P. Asente and R. Swick, *X Window System Toolkit, The Complete Programmer's Guide and Specification*. Digital Press, 1990.
- [19] N. Mayer, "WINTERP: An Object-Oriented Rapid Prototyping, Development and Delivery Environment for Building User-Customizable Applications with the OSF/Motif UI Toolkit.," Tech.Rep., Hewlett-Packard Laboratories, Palo Alto, 1991.
- [20] J. Ousterhout, "TCL: An Embeddable Command Language," in Proc. *1990 Winter USENIX Conf.*, (Dallas), pp. 133–146, 1990.
- [21] J. Ousterhout, "An X11 Toolkit Based on the TCL Language," in Proc. *1991 Winter USENIX Conf.*, (Dallas), pp. 105–115, 1991.
- [22] S. Selberherr, "Three-Dimensional Device Modeling with MINIMOS 5," in Proc. *VPAD 89 Workshop*, (Osaka), pp. 40–41, 1989.
- [23] J. Pfister, L. Parrillo, and F. Baker, "A Physical Model for Boron Penetration Through Thin Gate Oxides from p⁺ Polysilicon Gates," *IEEE Electron Device Lett.*, Vol. 11, No. 6, pp. 247–249, June 1990.
- [24] M. Lundstom and S. Datta, "Physical Device Simulation in a Shrinking World," *IEEE Circuits and Devices Mag.*, Vol. 6, No. 4, pp. 32–37, July 1990.
- [25] M. Orłowski, "Challenges for Process Modeling and Simulation in the 90's - an Industrial Perspective," in Proc. *SISDEP 91 Conf.*, (Zürich), pp. 3–22, 1991.
- [26] S. Laux, M. Fischetti, and D. Frank, "Monte-Carlo Analysis of Semiconductor Devices: The DAMOCLES Program," *IBM J.Res.Dev.*, Vol. 34, No. 4, pp. 466–494, July 1990.
- [27] J. Daniell and S. Director, "An Object-Oriented Approach to CAD Tool Control," *IEEE Trans.Computer-Aided Design*, Vol. 10, No. 6, pp. 698–713, June 1991.