Invited talk

Technology computer-aided design

Siegfried Selberherr
Technical University of Vienna, Institute for Microelectronics, Gußhausstraße 27–29, A–1040 Wien, Austria

Presented at STEDCON '92 16–18 November 1992

Computer simulation is an indispensable tool for the design of new VLSI devices. The demand for models capable of simulating both the processing steps involved in device fabrication and the electrical behaviour of devices is growing dramatically. The tight coupling of electrical device effects with the doping profile in device structures, as well as shrinking feature sizes are the main reasons. Models for the various fabrication processes and current transport in semiconductor devices are implemented as fairly complex software packages which have now served as simulation tools for engineers and researchers for more than a decade. Recent developments have demonstrated the need for integrating these process and device simulation tools into a common environment to allow centralized simulation control and information exchange. Such an environment is called a *technology computer-aided design* (TCAD) framework or system. Purely software-related aspects are attracting increasing attention. TCAD systems will aid device characterization and optimization considerably. As an example of this, the minimization of the substrate current of an *n*-channel MOS transistor is shown.

Rekenaar-simulasie is 'n noodsaaklike hulpmiddel by die ontwerp van nuwe VLSI-toestelle. Die aanvraag na geskikte modelle vir die simulasie van prosesseringstappe by toestelvervaardiging sowel as die simulasie van die elektriese werking van toestelle, neem tans geweldig toe. Hierdie toenemende aanvraag kan toegeskryf word aan die noue verband tussen die elektriese eienskappe van die toestel en die doteringsprofiele van die toestel se strukture, asook aan die toenemende pakkingsdigtheid. Modelle vir die onderskeie vervaardigingsprosesse, asook vir die stroomvloei in halfgeleiertoestelle, word as redelik ingewikkelde programmatuurpakkette geïmplimenteer. Ingenieurs en navorsers gebruik hierdie simulasiepakkette alreeds vir langer as 'n dekade. As gevolg van onlangse ontwikkelings het 'n behoefte ontstaan om die simulasie van die prosessering en die elektriese werking van toestelle te integreer in 'n gemeenskaplike omgewing om sodoende gesentraliseerde beheer van simulasie en die uitruil van inligting te bewerkstellig. So 'n omgewing staan bekend as 'n *tegnologie-rekenaargesteunde-ontwerp-* (TCAD-) stelsel. Belangstelling in aspekte wat uitsluitlik met programmatuur verband hou, neem toe. TCAD-stelsels sal veral nuttig wees by die ontwerp en optimering van toestelle. As toeligting word die minimering van die substraatstroom van 'n *n*-kanaal MOS-transistor beskou.

1. Introduction

Process and device simulation is commonly used for the design of new VLSI devices and processes. Simulation programs serve as exploratory tools in order to gain better understanding of process and device physics. On the other hand, simulations are also carried out after the design phase 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 coined. TCAD includes both physically rigorous as well as simplified process and device simulation in one to three spatial dimensions. Furthermore, links to layout-oriented CAD and circuit simulation are required.

Depending on the particular application of TCAD tools, different demands arise. On the one hand, the development of new technologies and the prediction of the behaviour of new devices require both accuracy and robustness. In this case, very sophisticated physical models and numerical techniques must be used, usually at high computational costs. 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 optimisation [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 modelling, 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 characteristics of the software which has been produced by that heterogeneous community during the past 20 years.

2. TCAD frameworks

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

2.1 TCAD versus ECAD

In the *electronic CAD* (ECAD) field, frameworks for tool integration and data interchange standards have emerged during the past years. Conversely, hardly any satisfying framework implementations are currently available for TCAD, especially for advanced, multidimensional process and device simulation. One reason for this is the difference in the sizes of the user communities. Electronic CAD is more widely used than Technology CAD.

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. 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 modelling there is no well-established unique data representation as in the case of ECAD.

These considerations indicate that the bottleneck in finding a unified data representation for TCAD is caused by semantic issues 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.

2.2 Framework demands

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

- allow minimum effort integration of existing tools
- 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 adapt easily to new design tasks
- provide an extensible database for design representation
- be "open" in terms of platform independence, availability and standards
- provide standard functionality such as visualization, interactive structure editing and postprocessing in the form of generic tools

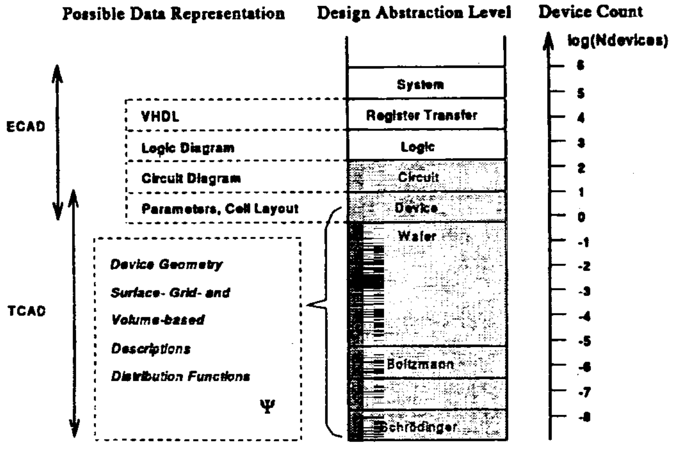

Figure 1 Technology CAD versus electronic CAD.

2.3 Existing approaches

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

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

Based on an initial proposal by Duvall [9], various Profile Interchange Format (PIF) based design environments both in industry (e.g. the PRIDE system [10]) and at universities (e.g. the PROSE system [11, 12, 13]) have been realized.

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

Depending on the intentions of the creators of the systems, different aspects, such as a rigorous task level implementation or a comfortable user interface [14], 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 [15] of the *CAD Framework Initiative* (CFI), an international standardization committee for ECAD. The intriguing goal of this approach is to separate the physical modelling 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 due 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 has been 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. The structure of VISTA is sketched in Figure 2.

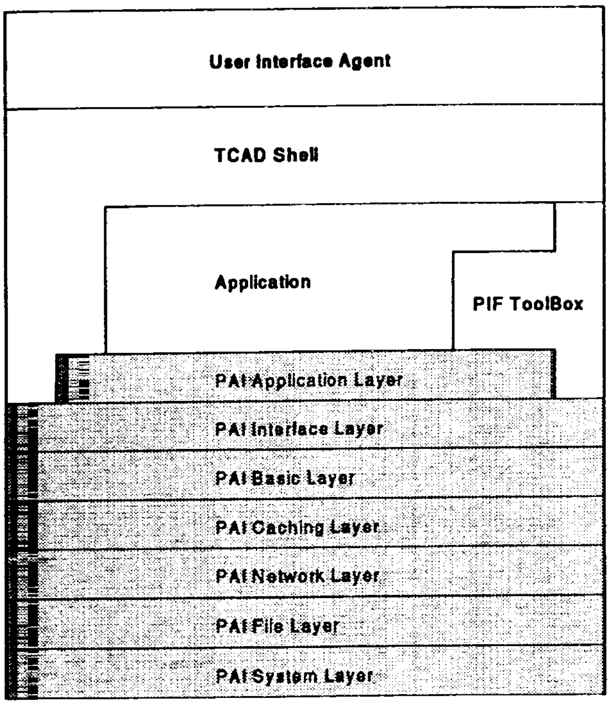

Figure 2 Layered Structure of the VISTA System.

Simulators and all other tools which we term applications, access the PIF database through the *PIF Application Interface* (PAI) [16], which supports several programming languages including C, FORTRAN and LISP. Thus, 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 much about the PIF syntax to be able to use the PIF database. In addition to full applications, PIF ToolBox functions are provided for generic data manipulation tasks like interpolation of physical data from one mesh to another.

The PAI was designed as a strictly layered product to guarantee the necessary functionality, performance and extensibility (cf. Figure 2). A *system layer* hides all system dependencies concerning communication with the operating system from the rest of the PAI. A *file layer* is used to handle all physical input/output requests. A *network layer* is responsible for functions on files in distributed computing environments. A *cacheing layer* takes care of performance and space requirements. The *basic layer* handles all structured data nodes and their relation. 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 [17] which integrates all system components on the task level. The major benefits of selecting LISP as extension language are that fully-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 [18] interpreter which was chosen as basis for the TCAD shell, is a publicly available software product written in portable C.

The user interface of VISTA [19] is based on the X Window system. By means of the X Toolkit [20] it implements most of the functionality which is required to support TCAD information flow using 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 [21, 22, 23].

An interactive device editor, generic postprocessing and visualization modules are also part of the system. 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.

3. An example

As an example, the bulk current of an *n*-channel MOS transistor has been minimized by varying the dose of the lightly doped drain (LDD) implant. Therefore process and device simulation are coupled within an optimization loop.

The doping profiles have been simulated with PROMIS, a two-dimensional process simulation tool [2]. The resulting transistors have been characterized by MINIMOS, a two- and three-dimensional FET analysis program [24]. Characteristic electrical parameters such as threshold voltage, saturation current, the maximum of the relative transconductance in the linear regime and the maximum of the bulk current have been computed in order to be controlled within certain prescribed intervals. The ratio of bulk to drain current for a constant bias condition is the value which drives the optimization loop for the LDD implant dose. The optimization has been controlled by the well-known golden section algorithm. The control flow diagram is depicted in Figure 3. Eight iterations were required to examine a dose range from 10^{10} to $10^{16}\,cm^{-2}$. That means eight process simulations with PROMIS and, due to the extensive device characterization, about 100 device simulations with MINIMOS were carried out automatically under the control of VISTA. As the resulting diagram of I_b/I_d versus LDD implant dose in Figure 4 shows, an explicit minimum exists and the device can be readily optimized. Needless to say, this task would not have been accomplished by mere experimental means; it would have been hard to carry it out using stand-alone simulation tools.

4. The future

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

The exact future development of physical models is hard to predict, as these always reflect the state of technology [25, 26]. Future models will have to be based on a lower level of

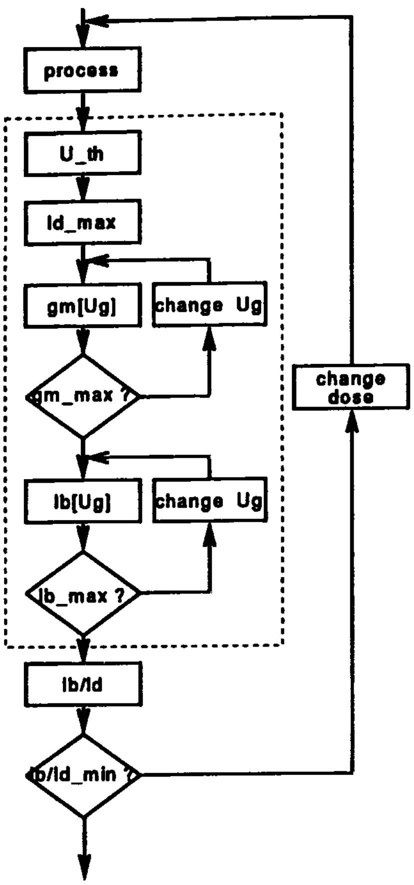

Figure 3 Control flow of a optimization loop.

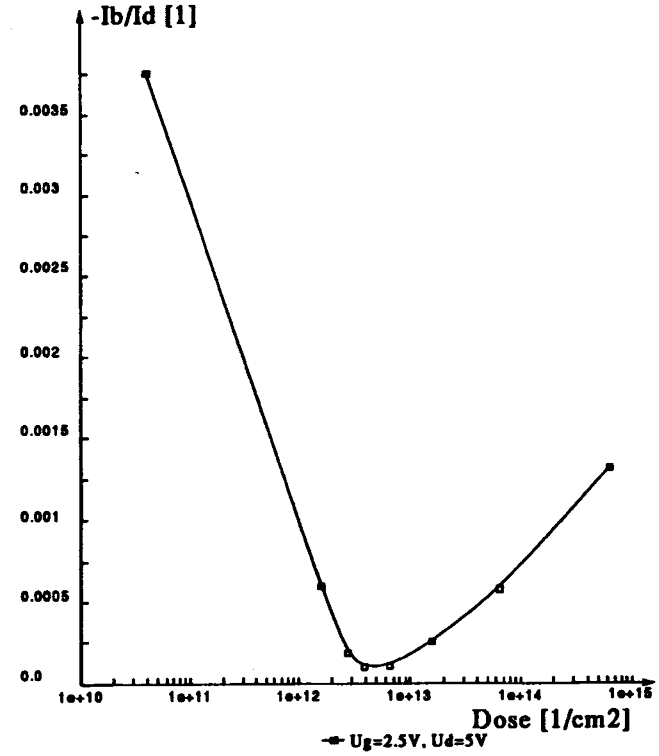

Figure 4 I_b/I_d versus LDD implanted dose.

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 [27].

Besides the progress in physical modelling, the future of process and device simulation will be significantly influenced by the introduction of TCAD systems and its impacts. This integration will allow simulation to catch up with the physical reality. It is to be hoped that TCAD systems will help to bridge the gap between the simulation needs of the engineer 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 of TCAD data which concern the contents and meaning of wafer and process information in a comprehensive and unambiguous way. 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 [28]. 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 technology computer-aided design 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, since not only inherent communication, data transfer and maintenance advantages exist but also new applications like global characterization and optimization are feasible with modern computers.

Acknowledgement

The VISTA project is supported by the "Forschungsförderungsfonds für die gewerbliche Wirtschaft", project 2/285, and by the research laboratories of Austrian Industries – AMS at Unterpremstätten, Austria; Digital Equipment at Hudson, USA; Siemens at Munich, FRG; and Sony at Atsugi, Japan.

References

[1] D.C. Cole, E.M. Buturla, S.S. Furkay, K. Varahramyan, 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, Solid-State Electron. **33** (1990) 591

[2] G. Hobler, S. Halama, W. Wimmer, S. Selberherr and H. Pötzl, Proc. NUPAD III (Honolulu, USA, June 1990) p.13

[3] J.B. Brockman and S.W. Director, Proc. NUPAD III (Honolulu, USA, June 1990) p.17

[4] J.S. Wenstrand, H. Iwai, M. Norishima, H. Tanimoto, T. Wada and R.W. Dutton, Proc. NUPAD III (Honolulu, USA, June 1990) p.15

[5] H. Matsuo, H. Masuda, S. Yamamoto and T. Toyabe, Proc. NUPAD III (Honolulu, USA, June 1990) p.59

[6] P.A. Gough, M.K. Johnson, P. Walker, and H. Hermans, IEEE Trans. Comput.-Aided Design **10** (1991) 808

[7] H. Jacobs, W. Hänsch, F. Hofmann, W. Jacobs, M. Paffrath, E. Rank, K. Steger and U. Weinert, Proc. NUPAD III (Honolulu, USA, June 1990) p.55

[8] P. Lloyd, H.K. Dirks, E.J. Prendergast and K. Singhal, IEEE Trans. Comput.-Aided Design **9** (1990) 1209

[9] S.G. Duvall, IEEE Trans. Comput.-Aided Design **7** (1988) 741

[10] M.R. Simpson, IEEE Trans. Comput.-Aided Design **10** (1991) 1163

[11] A.S. Wong and A.R. Neureuther, IEEE Trans. Comput.-Aided Design **10** (1991) 1157

[12] A.S. Wong, Technology Computer-Aided Design Frameworks and the PROSE Implementation, PhD thesis, University of California, Berkeley (1992)

[13] D.S. Boning, M.L. Heytens and A.S. Wong, IEEE Trans. Comput.-Aided Design **10** (1991) 1150

[14] E.W. Scheckler, A.S. Wong, R.H. Wang, G. Chin, J.R. Camanga, A.R. Neureuther and R.W. Dutton, IEEE Trans. Comput.-Aided Design **11** (1992) 911

[15] D. Boning, G. Chin, R. Cottle, W. Dietrich, S. Duvall, M. Giles, R. Harris, M. Karasick, N. Khalil, M. Law, M.J. McLennan, P.K. Mozumder, L. Nackman, S. Nassif, V.T. Rajan, D. Schröder, R. Tremain, D.M.H. Walker, R. Wang and A. Wong, Proc. NUPAD IV (Seattle, USA, June 1992) p.199

[16] F. Fasching, C. Fischer, S. Selberherr, H. Stippel, W. Tuppa and H. Read, Proc. SISDEP, ed. W. Fichtner and D. Aemmer (Zürich, Switzerland, September 1991) p.477

[17] H. Pimingstorfer, S. Halama, S. Selberherr, K. Wimmer and P. Verhas, Proc. SISDEP (Zürich, Switzerland, September 1991) p.409

[18] D.M. Betz, XLISP: An Object-Oriented LISP, Version 2.1 (Peterborough, U.S.A., 1989)

[19] S. Halama, F. Fasching, H. Pimingstorfer, W. Tuppa and S. Selberherr, Proc. NUPAD IV, ed. W. Fichtner and D. Aemmer (Seattle, USA, June 1992) p.237

[20] P.J. Asente and R.R. Swick, X Window System Toolkit, The Complete Programmer's Guide and Specification (Digital Press, U.S.A., 1990)

[21] 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)

[22] J.K. Ousterhout, Proc. 1990 Winter USENIX Conf. Proc. (Dallas, 1990) p.133

[23] J.K. Ousterhout, Proc. 1991 Winter USENIX Conf. Proc. (Dallas, 1991) p.105

[24] S. Selberherr, Proc. VPAD (Osaka, Japan, May 1989) p.40

[25] M. Lundstrom and S. Datta, IEEE Circuits Devices **6** (1990) 32

[26] M. Orlowski, Proc. SISDEP (Zurich, Switzerland, September 1991) ed. W. Fichtner and D. Aemmer, p.3

[27] S.E. Laux, M.V. Fischetti and D.J. Frank, IBM J. Res. Dev. **34** (1990) 466

[28] J. Daniell and S.W. Director, IEEE Trans. Comput.-Aided Design **10** (1991) 698