SIMULATION OF SEMICONDUCTOR DEVICES AND PROCESSES Vol. 3
Edited by G. Baccarani, M. Rudan - Bologna (Italy) September 26-28, 1988 - Tecnoprint

Technology CAD for Competitive Products

P. Lloyd, E.J. Prendergast and K. Singhal
AT&T Bell Laboratories
1247 South Cedar Crest Blvd.
Allentown, PA 18103, USA

Abstract

Recent trends in the integration of process, device and circuit modeling tools and the current rapid emergence of *UNIX®* based computing environments of networked workstations and compute engines makes possible user-friendly, task-based CAD systems for technology optimization, characterization and cell design. The paper discusses these trends and identifies opportunities to leverage Technology CAD tools in the development of competitive technologies and products.

1. Introduction

CAD tools for IC technology modeling are rapidly gaining maturity. Coupling the various modeling tools has been a recent trend particularly in industries where technology CAD has become an integral part of IC development [1-3] and is given organizational focus. The initial use of predictive CAD tools is generally as a substitute for physical experimentation to save time, effort and money, and to provide additional insight. In the second stage, tools are integrated and an optimization capability is added to evaluate competing technology alternatives in an automated manner. This is a valued capability as it is difficult for human intuition to work well in high dimensions and consider tradeoffs between different objectives that are often conflicting. In the next stage, it is recognized that the manufacturing process has inherent variability as do the operating and physical environments in which the product works. These variations cause product behavior to deviate from the designed nominal resulting in problems of yield and reliability. Traditionally, circuits have been designed by using a worst-case approach, often sacrificing either performance or yield. The real power of technology CAD tools lies in their effective use to make the production process and circuit design less sensitive to the inherent variations.

This paper describes how tools for technology CAD, in use at AT&T Bell Laboratories, are integrated into a task-based system designed to meet the needs of IC technology development and circuit design for optimization, characterization, and verification. TCAD is the name given to this system and it is made up of state-of-the-art, user-friendly modeling tools. These tools include those for process, device and circuit simulation, parameter extraction and optimization.

The TCAD environment is *UNIX*. The highest level of TCAD is a user interface program, *tcad*, written in C and shell. *tcad* is easily ported across *UNIX* systems so the investment in development is protected and efforts are not duplicated. It monitors access and usage patterns, tracks tasks routinely performed, has electronic communication facilities between users and program developers and provides various levels of restricted-access for certain sensitive or beta-stage tools and technology files. Provision is made to capture input files to add to the regression data-base, or when tools terminate abnormally. TCAD tools can be used independently or coupled together to form *tasks*.

The TCAD system is open to the integration of a variety of tools including exploratory products and the primary TCAD tools are accurate, flexible, robust and user-friendly. Some of these primary tools are described in the next section. The philosophy behind the TCAD architecture is outlined in the third section. Much of the power of TCAD comes from *UNIX* through networking and distributed computing. These features of *UNIX* and their impact on TCAD are described in Section 4. Several examples of TCAD tasks and concluding remarks are given in the fifth and sixth sections, respectively.

2. *The Building Blocks for TCAD*

TCAD leverages current predictive tools for process to cell level design. Some of these tools are described below including the process simulator BICEPS-5.0, the device simulator MEDUSA, the statistical process/device simulator FABRICS, the parameter extractor ARTHUR, the circuit simulator ADVICE and the optimizer CENTER.

2.1 The Process Simulator BICEPS-5.0

BICEPS-5.0 is a modular two-dimensional process simulator that retains the coordinate transformation philosophy of the original code [4]. It contains models for ion implantation, predeposition, drive-in in inert and oxidizing ambients, epitaxy, etching and deposition. The diffusion equation is solved using finite differences. Improvements over the original code include new physical models, an improved user interface, a new graphic package and new numerical methods.

The ion implantation module permits the use of Pearson IV and Gaussian impurity profiles. The modeling of a BF2 implantation has been introduced. For low energy boron implantation an additional distribution table, derived from measurements, is provided. A two-dimensional model for the out and in-diffusion of impurities into a poly layer has been introduced, permitting modeling of shallow emitters used in some bipolar technologies. Selective etching into silicon and selective epitaxy and

deposition are modeled.

At each fabrication step, BICEPS-5.0 automatically determines the grid necessary for accurate solutions and, where appropriate, extends the simulation domain laterally and horizontally. New numerical techniques have been introduced into BICEPS-5.0 to improve speed and accuracy.

2.2 The Device-Circuit Simulator MEDUSA

MEDUSA [5] is a user-oriented, mixed-mode, device-circuit simulator for the physical simulation of devices embedded in a circuit environment. Thus the influence of doping levels and geometry can be investigated, not only for single devices, but for subcircuits like inverters and memory cells. MEDUSA simulations are a sequence of DC analyses and/or transient analysis. Small-signal parameters of devices are extracted using Fourier-transforms.

The program provides a one-dimensional numerical model for bipolar structures. Input parameters are doping profiles and cross-sectional areas. Lateral voltage drops due to majority carrier flow, e.g. base resistance, can be taken into account by partitioning the device into several one-dimensional blocks that are coupled via transportances. These transportances are conductances modulated by the carrier densities in the incident one-dimensional subdevices. This quasi-multidimensional model is considerably more efficient than an exact three-dimensional analysis while preserving accuracy. MEDUSA also provides an efficient two-dimensional MOS model. SOI structures, transistors with non-planar surfaces, and transistors with up to two gates can be analyzed.

In addition to these predictive device models, MEDUSA includes compact models for diodes, bipolar transistors, and MOSFETs. An accurate table model is also available where the table can be populated using the numerical MOS model.

2.3 The statistical process/device simulator FABRICS

The FABRICS [6] simulator combines the functions of a process simulator such as BICEPS-5.0, and a device simulator such as MEDUSA. The key differences between FABRICS and these other simulators are:

(i) FABRICS relies on 2D models for the redistribution of impurities in Si. These models represent the solution to the partial differential equations governing the diffusion process under some simplifying assumptions (e.g. constant diffusivity). Models for oxidation, implantation and etching, are comparable to those found in other simulators.

(ii) In a similar fashion, FABRICS includes simplified device models that directly relate the device parameters (equivalent to the SPICE Levels 1 and 2 models) to physical quantities such as surface impurity concentrations, oxide thicknesses, and junction depths.

(iii) FABRICS runs three orders of magnitude faster than detailed numerical simulators such as BICEPS-5.0 and MEDUSA due to the simplicity of its models. Thus it is possible to carry out a statistical simulation by performing a Monte-

Carlo analysis using FABRICS when the sources of *noise* fall into two categories: (a) Noise in process control settings: i.e. assuming that process control parameters such as the times and temperatures of diffusion steps are subject to random fluctuations, and (b) Material variations: e.g. defect densities in wafers, which can be modeled as variations in the physical parameters governing the models, such as diffusivities and oxide growth rates.

To mimic the multi-level nature of noise in a fabrication line, where variations are observed between lots, between wafers within a lot, between chips on a wafer and between devices on a chip; FABRICS uses a multi-level random-number-generation scheme that associated means and variances with every level of the hierarchy.

(iv) To insure accuracy of simulated results, FABRICS is *tuned* to an existing process by identifying the set of physical parameters, such as diffusivities and oxide growth rates, in such a way as to have a *statistical match* between the observed process and the simulation results.

FABRICS is primarily intended for simulating MOS processes, and is being actively developed and extended at Carnegie Mellon University.

2.4 The general parameter extractor ARTHUR

ARTHUR is a model-independent parameter extraction system. It casts the parameter extraction task into a nonlinear least-squares minimization problem, and has an associated set of powerful interface tools that ease its application to arbitrary models. ARTHUR has been used to extract parameters for a variety of device models, including MOS, GaAs and BJT models, from measurements of current/voltage, capacitance/voltage and *s*-parameters.

Assume we have a model f of the form:

$$y = f(x,p)$$

where y is a vector of model *outputs* (e.g. currents), x is a vector of model *inputs* (e.g. voltages), and p is a vector of model parameters (e.g. V_{th}). The goal of parameter extraction can then be stated in the following way: given some measured output $\hat{y}$, and the corresponding input $\hat{x}$, determine the vector of parameters p^* which results in the *best* fit between the measurements and the model:

$$\underset{p}{minimize} \; || \, y - \hat{y} \, ||$$

where $||$ denotes the l_2 vector norm.

ARTHUR requires that, for a given vector of parameters p, there exist a mechanism to calculate the vector of residuals $\varepsilon = y - \hat{y}$. Thus, the most important architectural feature of ARTHUR is the *interface* between the extractor and the implementation of the model f. The interface relies heavily on the *UNIX pipe* facilities, as well as the CENTER/BRIDGE interface tools that allow for the

parameterization and control of general text files. This *building block* approach, whereby several tools are combined in a loosely coupled system, results in much greater flexibility than rigid single program subroutine-oriented systems. The ARTHUR architecture is shown in Figure 1.

The functions of some of the tools that form the CENTER/BRIDGE suite are: (a) **SPEC** enables the parameterization of arbitrary text files; (b) **MATH** extends the parameterization of SPEC to include mathematical constructs; (c) **LOGIC** allows conditional inclusion of portions of an arbitrary text file; and (d) **EGIL** is an interpretive programming language oriented to performing post-processing tasks. In such post-processing, we also make liberal use of *UNIX* tools like awk, sed, csplit and paste.

2.5 Circuit simulator ADVICE

ADVICE is a general circuit simulator with a large collection of models and simulation capabilities [7]. In terms of the TCAD task-based philosophy, ADVICE has three key features:

- *Procedural Simulation* [8]. The simulator may be completely controlled by writing a simulation driver in the C programming language. In addition to the control aspects, it is possible to access and manipulate the simulation outputs (e.g. voltage waveforms).
- *Modeling Extensions.* A tool named ADMIT [9] allows for extremely rapid introduction of new models into ADVICE. Specifically, the model need only be described in terms of its equivalent circuit, and the associated modeling equations for each element in the circuit. The ADMIT program will produce a customized version of ADVICE with the new model added.
- *User Interface.* State-of-the-art workstation-based user interface for simulator control and output analysis.

2.6 The Optimizer CENTER

CENTER is a generalized, numerical optimization system. *Generalized* means that it has a flexible interface so that, with the CENTER/BRIDGE tools, it can be easily hooked up to arbitrary design optimization problems. The flexibility allows CENTER to be integrated into TCAD as a module to perform high level tasks that require the execution of lower level TCAD tasks.

CENTER is a sophisticated optimization system. Disparate design optimization problems are often tackled best with different optimization algorithms, and CENTER contains many optimization algorithms. Optimization criteria are specified either as quantities to be minimized or maximized, or as design constraints (equality, upper and/or lower bound). CENTER is operated via a full-screen user interface that provides on-line help and does problem validation, uses dynamic memory allocation and embodies a simple, modular architecture by use of *UNIX* inter-process communication mechanisms. The overall system configuration for CENTER is shown in Figure 2.

A unique feature of CENTER is its ability to build mathematical models of design objectives and constraints as functions of design parameters. The optimization algorithms of CENTER can be applied to the models rather than the design problem, and the optimum predicted from the model validated against the actual problem. If necessary, the model is updated and the optimum prediction and validation procedure, based on the model, performed iteratively. This approach allows near-optimal designs to be found with substantially less computational effort than would be required for conventional optimization.

3. The TCAD Systems Architecture

The TCAD system allows experts to develop *tasks* for dealing with specific simulation situations. These tasks may be as general as the complete characterization of a technology as is done in the MECCA [10] system, or as specific as the evaluation of the latch-up susceptibility of a CMOS technology.

TCAD tasks are composed of three entities:

- *DATA* entities represent descriptions of the physical problem under consideration. Examples are fabrication process descriptions, profiles and layer thicknesses, current/voltage data and circuit performances.
- *TRANSFORMATIONS* relate the *DATA* entities. For example, process simulation relates a process description to a physical description of profiles and structure. Similarly, device simulation relates a physical description of a device to its electrical characteristics.
- *CONTROL* constructs determine a logical sequence of transformations.

In an abstract sense, a TCAD task is a directed graph (the *CONTROL*) where the nodes represent *DATA* and the arcs represent *TRANSFORMATIONS*. An example of such a directed graph is shown in Figure 3.

4. Hardware/Software Issues in the Implementation of TCAD

The dramatic advance in CPU performance and high-speed networking hardware has been rivaled only by the drop in their respective costs. Computing has emerged from a multiple user/single CPU environment (as in traditional time sharing systems) and one-to-one user/CPU environments (personal computers, workstations, etc.), to a distributed computing environment where multiple CPUs are at the disposal of a single user. The distributed computing model provides open computing through standardization, network wide shared information, most effective cost ratio (MIP/dollar), and graphically oriented and individualized computing environments.

An important feature of this *next generation* of computing has been the adoption of international and defacto networking and operating system standards. The rapid growth of computer network technology has led to unprecedented interconnectivity among machines manufactured by different vendors. A key feature of the growth has been the selection of a standard operating environment, the *UNIX* operating

system. The *UNIX* operating system has historically featured the more advanced features of computer networking (TCP/IP, remote logins, remote shells, etc.). The introduction of the Network File System (NFS) has provided transparent remote file access. Users and applications can access files without acknowledging on which machine the files are located. Remote communication protocols have been standardized across *UNIX* systems enabling networking among different architectures. *UNIX* runs on a multitude of hardware platforms manufactured by different vendors (SUN, Cray, DEC, Amdahl, Alliant, etc).

In a distributed computing system, the combined resources of the network become accessible to each user. It is possible to tune applications to the availability of the network's resources. For example, a floating point intensive application may be off-loaded to a special-purpose number crunching computer for maximum performance. More typically, an application may be broken down into autonomous components which may be distributed and run asynchronously over the network. Since all machines on the network can access the networks file systems, each component task can share information transparently.

For TCAD, a distributed computing network offers many advantages. The umbrella model employed by TCAD is very much like the computer network. The loose coupling of tasks within a TCAD application makes it highly suitable for distributed processing. TCAD tools have a consistent, generalized interface for data communication. The use of such interfaces allows modules to be linked and reused easily, and the adoption of a consistent protocol for data communication makes addition of new modules to the system simple. High quality workstations provide superior user interfaces. The CPU intensive tasks are distributed to more powerful multiprocessor machines. While each task of an application is graphically monitored, other types of applications may be started or continued.

5. Examples

5.1 The TCAD-MECCA Design-Thread

The MECCA system and approach to integrated modeling has been in existence since the early 1980's [1]. MECCA operates in both a technology characterization mode and an optimization mode. It is based on the use of process and device simulation tools, parameter extractors and a general optimization package. In the characterization mode MECCA generates nominal and worst-case parameter files for circuit simulation including temperature parameters and capacitance values. In the optimization mode MECCA simplifies the accurate comparison of significantly different technology alternatives since all options can be optimized in terms of a fixed set of technology objectives. MECCA is a design-thread under TCAD and a flowchart is shown in Figure 4.

5.1.1 MECCA for Technology Characterization: MECCA (Characterization) is the procedure used to generate parameter files for use in the circuit simulator ADVICE. If the technology is CMOS, for example, ASIM [11] (Figure 5) and CSIM [12] parameter files are generated. As seen in Figure 4, the work horses of

this system are BICEPS-5.0, for process simulation, MEDUSA, for device simulation, and a variety of parameter extraction and data reduction tools, such as ASIMPAC [11], MOSPAC [12], ARTHUR and PAREX [13]. Given a technology specification and process sequence, BICEPS-5.0 is used to generate one or two-dimensional doping profiles for the devices or structures being characterized. Profiles for nominal and extreme cases are obtained by varying implant doses, drive-in times and temperatures consistently with expected variations in the fabrication line. Extreme case files include not only fast and slow transconductance parameters but also those associated with complex mismatch situations, essential for aggressive linear designs. These profiles, along with geometry specifications, physical constants or parameters and grid information are used as input to MEDUSA to generate the electrical characteristics required for parameter extraction. Transient simulations generate data for the extraction of capacitances and ac/frequency parameters. All these simulations are then repeated for the extreme cases. These techniques provide a predictive capability for generating files. Files are verified with device measurements as wafers become available. If a change is deemed necessary, a detailed investigation is carried out to determine and correct weaknesses in the various modeling tools and hence improve confidence in predictions for the next generation of technology.

Recent work has been focused on the complete automation of the MECCA (Characterization) system under the TCAD framework. In the *UNIX* environment, tools such as awk, sed, grep, m4 and shell programming have been valuable aids in accomplishing this task.

5.1.2 MECCA for Technology Optimization: Figure 4 shows that the input to MECCA (Optimization) is an initial guess for a process specification along with constraints on the performances to be optimized. The output is a process specification including device geometries, optimized to meet a set of technology objectives, that is, parameters to be maximized, minimized or constrained.

MECCA (Optimization) operates as follows: The initial guess at a process description is simulated by BICEPS-5.0 generating impurity profiles for MEDUSA. In turn, MEDUSA generates a set of variables that characterizes a technology including the performances to be optimized. The optimizer CENTER generates new input for BICEPS-5.0 and MEDUSA in an attempt to meet the technology objectives. The optimization loop is broken when the technology objectives are achieved to within a specified tolerance.

An example of the use of MECCA (Optimization) for the optimization of the drive/speed of a CMOS technology is given elsewhere [10]. In it's original form, MECCA (Optimization) was a single FORTRAN program with information passed via subroutines. As a design thread in TCAD, the system has become modular, flexible and user friendly with only minor performance penalty. A recent application of MECCA (Optimization) was to the drain engineering of a CMOS technology to minimize *hot* carrier generation. Peak substrate current, calculated using a non-local impact ionization model in MEDUSA, was used as a measure of the generation term to be minimized. Accurate substrate current modeling requires

a full solution of energy transport [14], however, simple drift/diffusion is adequate for showing trends particularly in the sensitive region of high drain voltages and low gate voltages. The result of the optimization was a reduction in substrate current by 20% without significantly affecting device performance.

5.2 TCAD Automated Interconnect Modeling

As circuit speeds increase and device dimensions decrease, the interconnect plays a greater role in the behavior of integrated circuits. A TCAD task is the characterization of the interconnect systems in a technology for simulation purposes. This would include local (contacts etc...) as well as global (transmission-line) features. Here we describe the modeling of the line to ground and interline capacitance for a technology including the statistical variations.

The method is to use detailed numerical simulation to generate tables of capacitance vs. lateral (conductor widths/separation) and vertical (layer thickness) dimensions. As the numerical simulation would be too expensive for each individual geometry, general model building tools are used to generate simple analytical expressions suitable for use in circuit extraction programs.

As an example, consider the characterization of a typical CMOS technology. The technology specification provide suitable limits on the vertical and lateral dimensions. An efficient experiment plan in those dimensions is then formed. The two-dimensional Laplace equation solver RESCAL [15] is used to calculate the capacitance of a single conductor, a pair of conductors, and three conductors over the field oxide region.

The capacitance/dimension data is fed to a multi-nomial regression program that produces C-code for models for direct inclusion into the circuit extractor. Figure 6 shows the overall interconnect capacitance computation task thread. A TCAD task is thus a system for substantially automatically generating the required interconnect models from technology descriptions.

5.3 The TCAD-CENTER Design Thread

Process, device and circuit simulation tools provide the capability to predict the behavior of devices and cells. The next step in TCAD tool integration is to optimize the technology and circuits to enhance performance and reduce sensitivity to process and environmental variations. The software architecture of CENTER simplifies the integration of the predictive tools with optimization algorithms thus providing a powerful design environment. We illustrate the design optimization capability through simple examples.

5.3.1 Design of a Digital PLL: The design parameters for a digital phase lock loop (PLL) were the lengths (number of bits) of an up/down counter (UDC) and of a random walk filter (RWF). The design objectives were the minimization of the lengths of the counter and filter with a design constraint on the jitter.

The jitter for the PLL with initial UDC and RWF lengths violated the specification. Application of CENTER to the problem yielded a design that

satisfied the design constraint with a combined UDC and RWF length smaller than that of the original design.

5.3.2 Robust (statistical) Design example: CMOS Voltage Reference: Integrated circuit manufacturing processes exhibit statistical variations between fabrication of different lots, and the electrical and ambient conditions under which circuits are operated vary. Consequently, the performance of specific realizations of a circuit may differ from that of the specified nominal behavior. Good circuit design practice should therefore be directed towards generating circuit designs that exhibit the best performance characteristics in a statistical sense rather than just for a specified nominal case. We illustrate this principle on the simple CMOS voltage reference, Figure 7.

The design parameters were the transistor widths. The design objective was to minimize the mean squared deviation of the output voltage from the specified nominal value of 2.5 volts for various circuit processing and operating conditions. Variations in 7 processing parameters, the supply voltage and load current were to be accounted for. CENTER was applied to carry out the high level optimal statistical design task, FABRICS was used for the combined tasks of process simulation, device simulation and technology characterization and SPICE was used to perform circuit simulation. The flow of data between tasks was effected using both the CENTER/BRIDGE tools and the *UNIX* program awk for generalized data transformations. The overall task thread is shown in Figure 8.

50 statistical samples of the process were generated and 3 values (low, nominal and high) of the supply voltage and load current, resulting in 450 SPICE circuit simulations, were used for the statistical characterization of each design. The computational expense of each design characterization dictated that efficient optimization techniques be used. CENTER can generate a sparse sample over the design space and build internal models of the objectives as functions of the design variables. Optimization is performed, at relatively little computational cost, on an internal model and the prediction compared with the actual design. If a large discrepancy exists, the model is updated and and the process continued. The application of robust design principles led to an increased yield from 27% to 35% of circuits whose output reference voltage was within the range of 2.4 to 2.6 volts under all operating conditions.

The example shows that circuit design can be improved if statistical process and operating condition variations are taken into account during the design process, rather than just performing a nominal design. However, the computational cost of such design is substantial and requires the application of efficient optimization techniques, such as the new approaches available in CENTER.

5.3.3 Example: Statistical design of an analog filter: As another example of design for manufacture, consider the case of the analog filter shown in Figure 9 which needs to meet the specifications shown in Figure 10. Owing to the inherent tolerances on the components, some of the manufactured filters fail to meet specifications. The yield can be enhanced by purchasing tighter toleranced

components at higher cost. Often, designers do *worst case* design and specify tight tolerances to get 100% parametric yield. However, it has been found that the minimum cost solution to the problem often lies at a yield somewhat below 100% as tolerances can be substantially reduced.

In the example shown, let us take the normalized component cost to be $1/t_i$ for the capacitors and $2/t_i$ for the inductors. A fixed cost of 2.5 units represents assembly cost. Filters that fail specifications are discarded. The objective then is to minimize the expected cost per satisfactory circuit:

expected cost = (Component cost + fixed cost)/yield

by selecting the component nominals and tolerances. While component cost is directly related to the tolerances, the yield is a complex function of the design variables and, in general, can only be found by Monte Carlo analysis. A direct approach to this design optimization problem is not feasible and over the last two decades many researchers have proposed novel methods for its solution. One approach that has proved effective is based on *Parametric Sampling* that makes effective sample re-use as the optimization proceeds [16]. In the above example, the normalized cost was reduced from 4.53 for a worst case design to 3.32 for a statistical design with a yield of 97%. This reduction in cost was achieved in only 160 circuit analyses.

6. *Conclusion*

Technology CAD is being established as a discipline and given organizational focus in the IC industry. At AT&T Bell Laboratories, our TCAD system provides us with rapid integration and deployment of tools and tasks essential to process through cell level characterization and design. The ability to provide design centering and extreme case analysis related to process variables has been a key to aggressive design of robust IC products. This capability results from i) efforts in the early to mid 1980's to establish integrated modeling systems such as MECCA ii) the application of general purpose optimization capabilities such as WATOPT [17] and CENTER iii) recent advances in open and powerful computing environments (principally *UNIX* based distributed resources). The open nature of current systems for Technology CAD and the trend towards standards [18] offers much promise as a basis for collaborations within companies and for effective integration of modular software products from universities.

7. *Acknowledgments*

We gratefully acknowledge the work of many collaborators and colleagues at AT&T Bell Laboratories, the Universities of Aachen and Waterloo and Carnegie Mellon University who have contributed to the work described in this paper.

8. *References*

[1] P. Lloyd: Application of Numerical Simulation in Modeling of IC Device Structures. Proc. NASCODE III, Galway, 1983.

[2] R.W. Knepper, S.P. Gaur, F.Y. Chang and G.R. Srinivasan: Advanced

Bipolar Transistor Modeling: Process and Device Simulation Tools for Today's Technology. IBM J. Research and Development, No. 3, May 1985.

[3] J. Mar, K. Bhargavan, S.G. Duvall, R. Firestone, D.J. Lucey, S.N. Nandgaonkar, S. Wu, K. Yu and F. Zarbakhsh: EASE - An Application-Based CAD System for Process Design. IEEE Trans. on CAD, November 1987.

[4] B.R. Penumalli: A Comprehensive Two-Dimensional VLSI Process Simulation Program, BICEPS. IEEE Trans Elec. Dev., Sept. 1983.

[5] W.L. Engl, R. Laur, H.K. Dirks: MEDUSA - A Simulator for Modular Circuits. IEEE Trans. CAD, April 1982.

[6] S.R. Nassif, A.J. Strojwas and S.W. Director: FABRICS II: A Statistically based IC Fabrication Process Simulator. IEEE Trans. CAD, pp. 20-46, January 1984.

[7] L.W. Nagel: ADVICE for Circuit Simulation. Proc ISCAS, Houston, 1980.

[8] T.F. Fang et al: ADVsh: Procedural Circuit Simulation and Simulation Control. Submitted for publication.

[9] S. Liu, K.C. Hsu and P. Subramaniam: ADMIT-ADVICE Modeling Interface Tool. 1988 Custom Integrated Circuits Conference, Rochester, 1988.

[10] E.J. Prendergast: An Integrated Approach to Modeling. Proc. NASCODE IV, Dublin, June 1985.

[11] S.W. Lee and R.C. Rennick: A Compact IGFET Model - ASIM. To be published in IEEE Trans. CAD.

[12] S. Liu and L.W. Nagel: Small Signal MOSFET Models for Analog Circuit Design. IEEE J. Solid State Circuits, December 1982.

[13] G.M. Kull, L.W. Nagel, S-W. Lee, P. Lloyd, E.J. Prendergast, H.K. Dirks: A Unified Circuit Model for Bipolar Transistors Including Quasi-Saturation Effects. IEEE Trans Elec. Dev., June 1985.

[14] B. Meinerzhagen: Two Dimensional Numerical Substrate Current Modeling for n-channel MOS Transistors. Proc. NASCODE V, Dublin, June 1987.

[15] B.R. Chawla and H.K. Gummel: A Boundary Technique for Calculation of Distributed Resistance. IEEE Trans. Elec. Dev., October 1970.

[16] K. Singhal and J.F. Pinel: Statistical Design Centering and Tolerancing using Parametric Sampling. IEEE Trans. CAS, pp. 692-702, July 1981.

[17] R. Chadha, K. Singhal, J. Vlach, E. Christen and M. Vlach: WATOPT-An Optimizer for Circuit Applications. IEEE Trans. CAD, pp. 472-479, May 1987.

[18] S.G. Duvall: An Interchange format for Process and Device Simulation. IEEE Trans. CAD, pp. 741-754, July 1988.

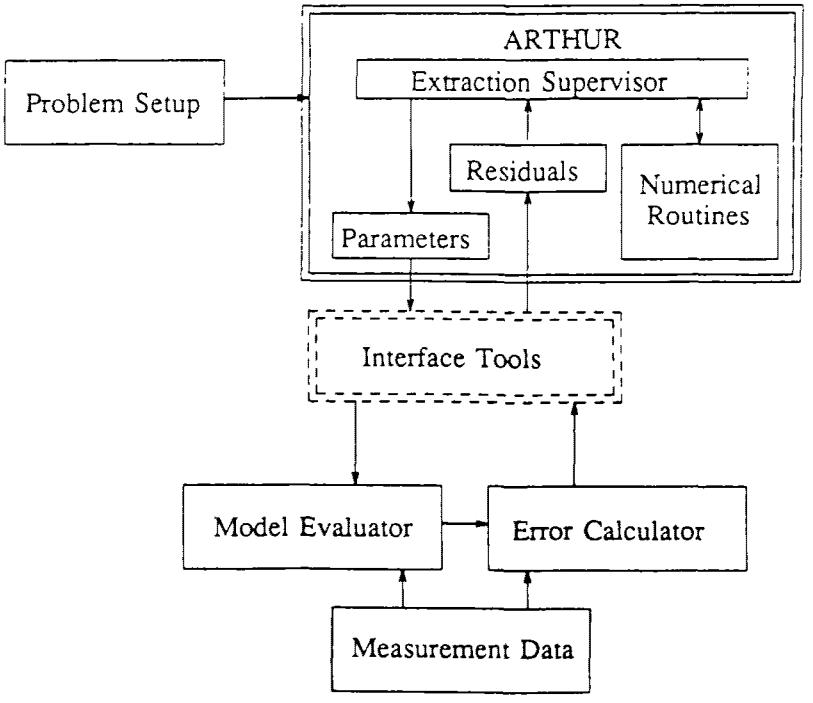

Figure 1. The ARTHUR parameter extractor

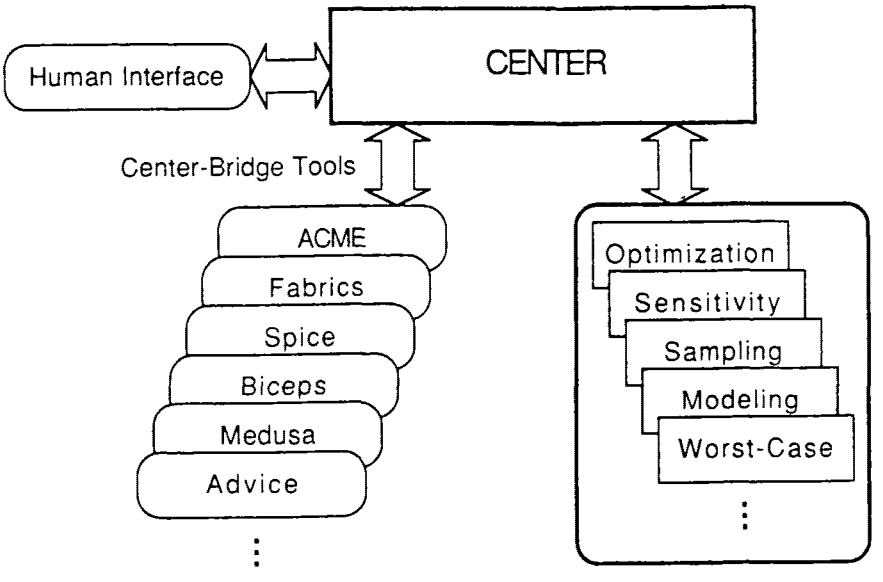

Figure 2. The optimization system CENTER

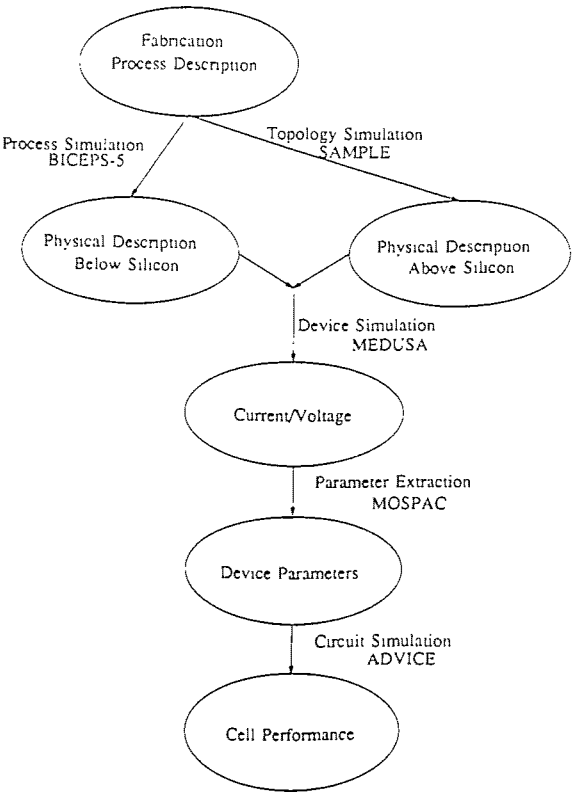

Figure 3. Example of a TCAD task

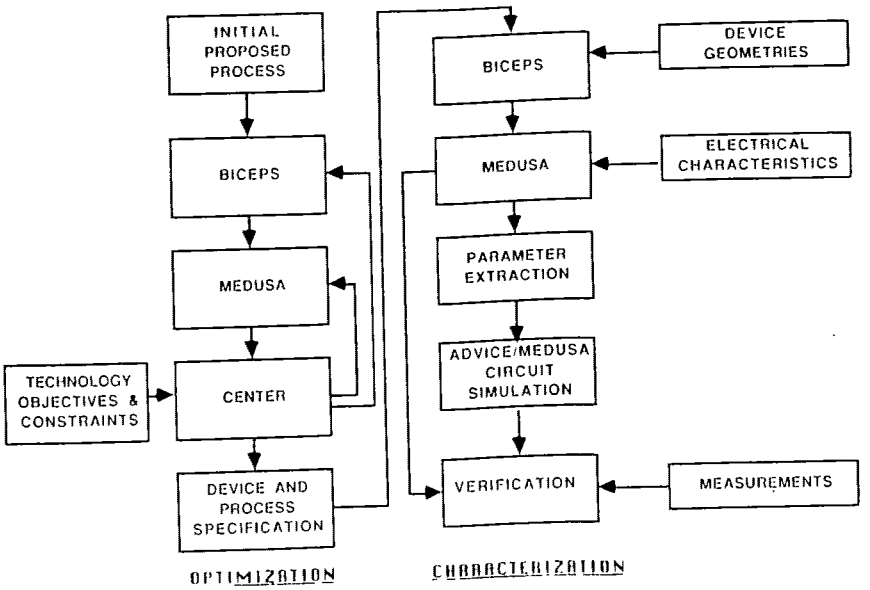

Figure 4. The MECCA optimization/characterization system

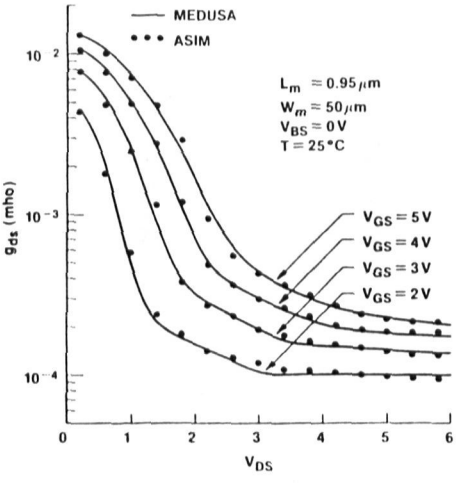

Figure 5. Output conductance fit from ASIM

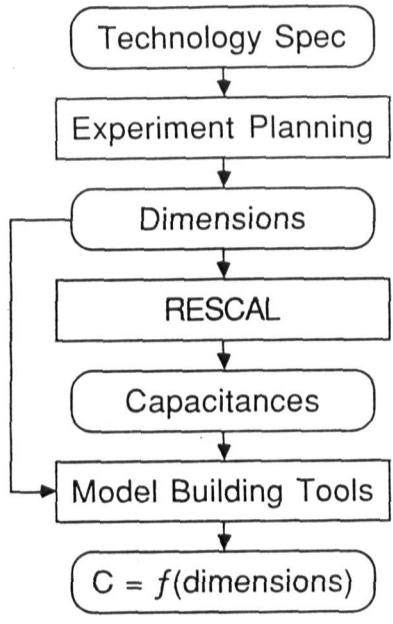

Figure 6. interconnect capacitance task thread

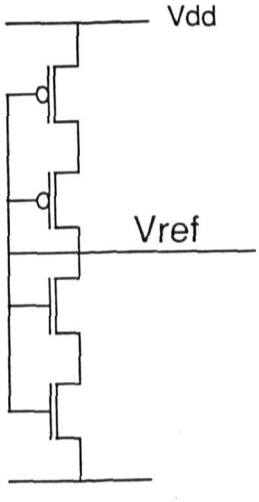

Figure 7. Voltage reference circuit

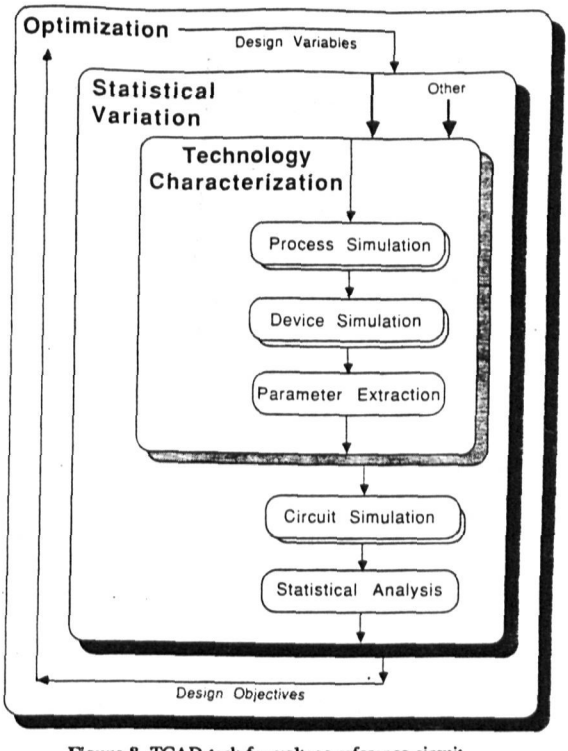

Figure 8. TCAD task for voltage reference circuit

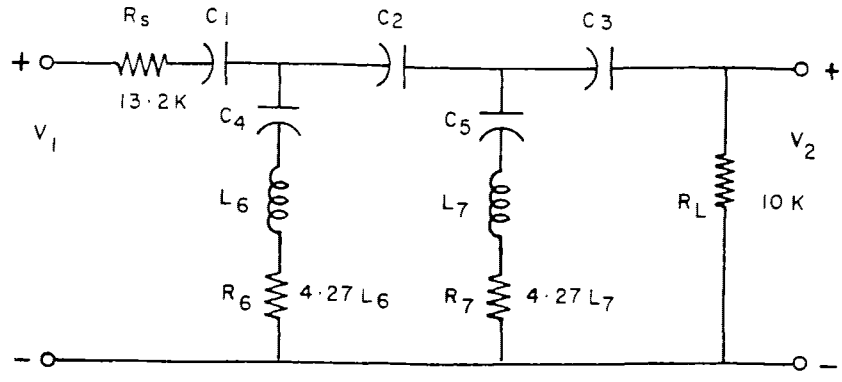

Figure 9. Analog filter circuit

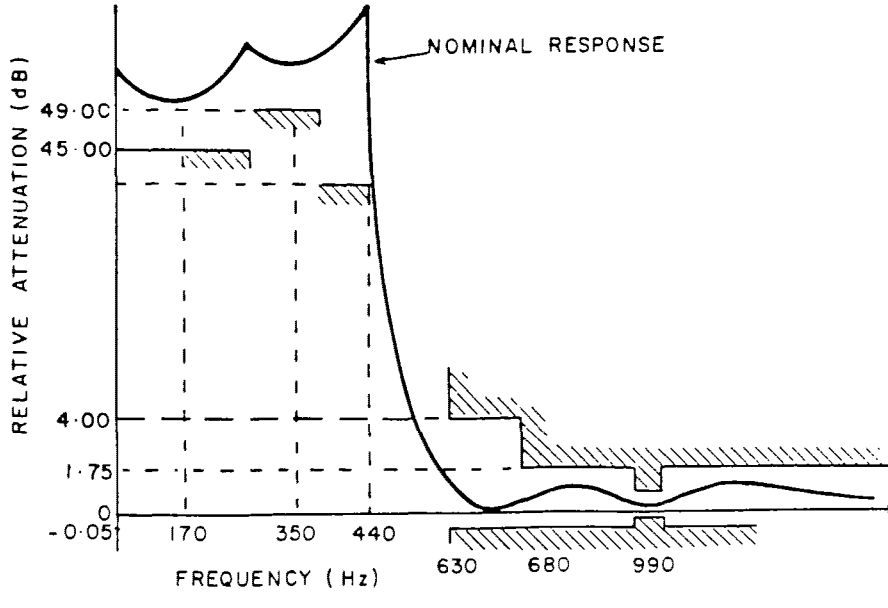

Figure 10. Analog filter performance specifications