The Control System of the Device Simulator MINIMOS-NT

Robert Klima, Tibor Grasser, and Siegfried Selberherr
Institut für Mikroelektronik
Technische Universität Wien
Gußhausstraße 27–29, A-1040 Wien, Austria
Klima@iue.tuwien.ac.at http://www.iue.tuwien.ac.at

Abstract: - State-of-the-art TCAD software like the multi-dimensional device and circuit simulator MINIMOS-NT require a huge amount of information in addition to the device input data to control several different complex modules and tasks. This information is normally hierarchically structured containing arbitrary cross-relations and dependencies that include, e.g., material properties, circuit descriptions, models and model parameters, or simulation parameters. Therefore, the control system for TCAD software must be able to handle these data and to allow simulation control in an efficient and comfortable manner. To obtain a maximum of flexibility and controllability a new specialized object-oriented dynamic database is used.

Keywords: - Systems and Control, Object-Oriented Programming, Simulation Control, TCAD Software, Software Development, Object-Oriented Dynamic Database

1 Introduction

Device simulation has emerged as a flexible and powerful tool to aid design and optimization of semiconductor devices. State-of-the-art three-dimensional device simulators like MINIMOS-NT [1] are complex software tools and challenging to control. The simulated device is usually loaded from a file containing the geometric definition, the grid, and the input quantities defined on the grid. Nevertheless, this information is not sufficient. To perform a simulation additional information is needed to describe the device, its segments, and an eventually connected circuit. Moreover, adjustment of models, model parameters, material properties, circuit descriptions, simulation modes, iteration schemes, or input/output definitions must be possible. Thus, a sophisticated control system is required to allow control of each part of the simulator.

Nevertheless, even for such complex control systems the user-interface must be easy to learn and the user must be able to set up the specification files to describe a simulation in a straight forward and flexible way. The types of the additional information needed for device simulation are usually manifold:

- Values are the simplest kind of information. They may be, e.g., model parameters, quantities, contact values, limits, or just names and strings.
- Expressions are mandatory to describe dependencies between different parameters. These expressions may be mathematical formulas or logical relations and conditions.
- Sequences describe the order in which a given set of elements is processed.
- Hierarchies map the structures of hierarchical information. Hierarchies are used on the one hand side to describe relations between structures and on the other hand side to define the contents of a structure.

In the past years an object-oriented dynamic database called Input Deck database [2, 3] has been developed to meet the requirements of state-of-the-art TCAD applications. This database has been used to control several TCAD tools like the device simulator MINIMOS-NT, the Wafer State Library [4, 5], and the simulator FEDOS (Finite Element Diffusion and Oxidation Simulator) [6]. In this work we present the complex control-system of the device-simulator MINIMOS-NT.

2 The Input Deck Database

The Input Deck database [2, 3] is a new kind of database and works rather differently compared to common databases. The most important items stored are variables and sections. Variables queried by the simulator are called keywords. Variables do not act like variables known from common programming languages. In fact, they represent states and act similar to cells known from spreadsheet tools. Variables contain expressions to describe dependencies to other variables. Sections hold an arbitrary set of variables and an arbitrary number of nested subsections to build any desired kind of hierarchies.

Expressions are evaluated at runtime when a value of a variable is queried by the simulator. Inside the database expressions are stored very efficiently as a special kind of forest [2] which enables both low memory consumption and a very fast evaluation of expressions.

The main advantage of the dynamic evaluation during runtime is that the values of all variables change virtually when the value of a variable they refer to is modified. Virtually means that recalculation of depending variables only happens when their values are queried. Thus, no calculation overhead occurs when values are changed by the simulator. Another feature of the Input Deck database and the control mechanism of MINIMOS-NT is that the simulator is allowed to change certain variables which are marked writable, thus feedback loops are possible (see Section 8).

The Input Deck uses a powerful inheritance scheme [2] called explicit-inheritance scheme to pass complex hierarchies of nested sections and their variables to other sections. This approach is used, e.g., for the material database of MINIMOS-NT (see Section 5), for the definition of circuits (see Section 6), and for the defaulting system of the simulator (see Section 4).

The Input Deck database has its own database description language, the Input Deck description language [1] (IDL). The syntax of the IDL has been defined similarly to that of the C++ programming language because C++ is one of the most popular programming languages. Nevertheless, the IDL is not a sequential programming language as it describes the entries of a database.

The Input Deck database offers an extensive application interface (API). The simulator can query variables and sections by their name or by means of iterators. Iterators are well suited to explore hierarchies and sequences. Application specific functions like the stepping functions (see Section 7) can be applied to extend the functionality of the database.

The default settings of MINIMOS-NT consist of approximately $2,400$ keywords and $1,000$ sections defined in the MINIMOS-NT default files. Reading and parsing of these files requires $20ms$ CPU time on a 1GHz Pentium Linux computer. Due to inheritance approximately $30,000$ keywords and about 8.000 sections exist. Evaluation of all keywords requires $1.5s$.

3 The Simulation Flow

MINIMOS-NT is controlled by the Input Deck database in each phase of the simulation flow. Depending on the phase of the simulation the database performs different tasks. For instance, simulation modes, iteration schemes, or parameters for the solving process have to be chosen. Since state-of-the-art device simulators are very complex a multitude of decisions are imaginable for the user to influence the simulation flow.

Fig. 1 shows a schematic outline of the structure of the device simulator MINIMOS-NT. Each phase describes a complex process which must be controlled. The dashed arrow-lines indicate the direction of the communication with the Input Deck database.

At the beginning of the initialization phase the Input Deck database is initialized. After the initialization of the database, MINIMOS-NT is controlled by means of the database. The basic structures and all modules and libraries are initialized, the device and circuit definitions and additional information like material properties (see Section 5) are loaded.

In the next step the device or the circuit and its embedded devices are loaded and analyzed. For each segment of each device the material is determined, the corresponding definitions are loaded from the database, and the input quantities are loaded. Contact definitions are read from the Input Deck database. Depending on the simulation mode, the segment material, and the models to use which are set in the database the simulation quantities are initialized. Initial quantities like the interface distances or the net impurity concentration are calculated.

MINIMOS-NT provides a powerful stepping mechanism (see Section 7). This stepping allows to step arbitrary parameters and enables to run simulations for all permutations of a given set of parameters.

Within the main simulation loop three important phases are processed. In the preprocessing step the necessary quantities are calculated either from the initial state, the previous iteration, or the previous simulation point.

In the solving step the linear equation system is assembled. The iteration scheme and the solving proce-

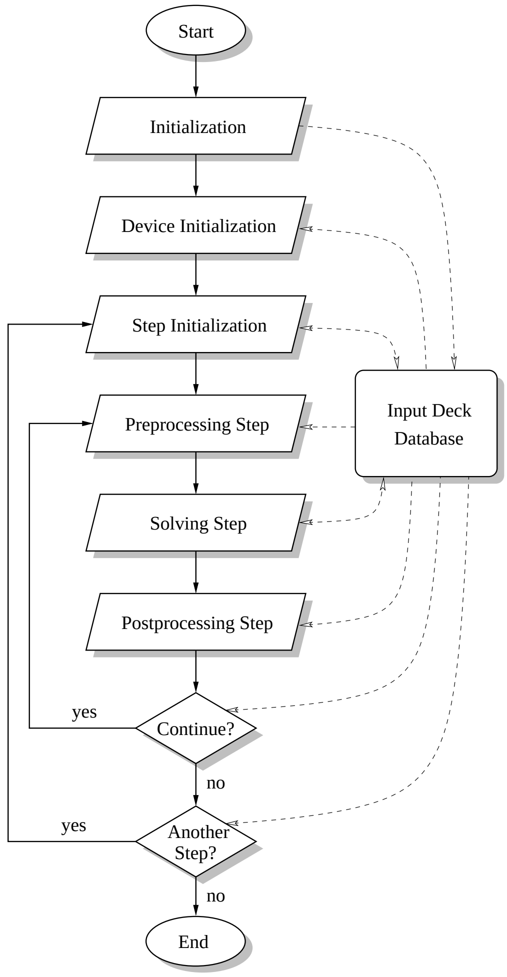

Figure 1: Schematic structure of a device simulator.

dure (see Section 9) are controlled by the Input Deck database. Runtime information on the state of the solving process is written back to the database which causes other expressions to change (see Section 8).

In the postprocessing step the output quantities are calculated from the computed solution. Moreover, several decisions are imaginable to influence the simulation loops (in Fig. 1 only the outer simulation loops are shown). Some of these approaches are depicted in Section 9.

4 Defaulting System

To provide default values for all possible keywords, MINIMOS-NT extensively uses the inheritance scheme of the Input Deck database. The default values are stored in separate sections. By convention, the names of these sections end with `Defaults`. For instance, the default iteration schemes are defined in the section `IterateDefaults`, the default settings for devices are stored in the section `DeviceDefaults`.

To use the default values the corresponding global default sections are inherited to the sections needed by the simulator, the standard sections of MINIMOS-NT. For instance, to use the standard settings for a device, the `Device` section is inherited from the `DeviceDefaults` section which holds all the default information. Also, those sections and the default inheritance schemes are predefined.

To modify the settings, the user can simply change the standard sections by locally overriding the default values. The Input Deck database assists the user by applying a checking mechanism when local modifications are performed in sections. Only those items defined in the parent sections which are inherited to the standard sections may be overloaded. Otherwise an error message is thrown to avoid misspelling of names. New items can be added but they have to be marked explicitly. If this checking mechanism is not desired, it can be turned off for each section.

5 Material Database

MINIMOS-NT uses a variety of physical models which are managed by a separate module of the simulator, the model server [7]. This model server allows the user to define customized models. Since the models to use depend on the type of the material, efficient material handling becomes an important issue. MINIMOS-NT uses only abstracted classes of the material, which are `Semiconductor`, `Insulator`, `Conductor`, and `IdealConductor`. The properties of the material classes are described via a model set, for instance the mobility or relaxation times of semiconductors. These models are defined as black boxes using a set of input and output parameters. Each black box is replaced by a certain implementation of this model which calculates the output values. Several implementations are available, each with the same set of model-parameters.

The material database manages all materials known by the simulator. The materials are hierarchically structured (see Fig. 2). To handle common material properties, real materials like Si or SiO_2 are inherited from abstract materials like semiconductor and insulator. Later on, the material properties of the derived material classes are extended and specialized.

Each material is represented by a section which provides the parameter values for all available physical models which are hierarchically structured. For

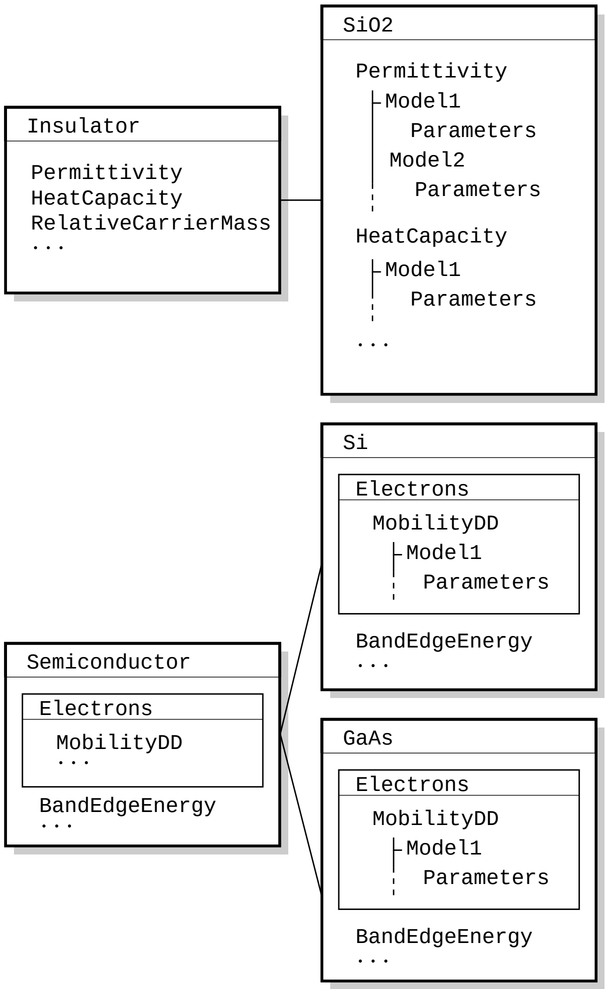

Figure 2: Material Database.

each model class (e.g. the mass density or the band edge energy) and for each material (e.g. Si or SiGe) certain model instances can be chosen with so called model-specializer keywords.

For each geometric region in a device all physical models to use and their parameters must be specified in a special section. By inheriting the global material database the default settings are loaded which can be locally overridden.

The advantages of this hierarchical approach are:

- As MINIMOS-NT only deals with abstract material classes new materials can easily be added by inheriting them from the existing material classes.
- New models can be easily tested and calibrated by overriding the default values for these models.
- The material and model defaults can be easily customized by changing or adding new entries in the global material database.
- Maintenance of the default settings is much easier compared to hard coded default values in the simulator.

6 Mixed-Mode

MINIMOS-NT has been equipped with circuit simulation capabilities [8]. The circuit may consist of an arbitrary number of compact and distributed devices. Default compact devices are, e.g., several kind of sources, resistances, conductors, capacitors, inductors, diodes, or transistors. Any number of distributed devices may be added to the circuit, each of them described by a separate section in the database. Instead of applying voltages or currents at the contacts as it is the case for single mode simulations the contacts are connected to nodes in the circuit. Thus, the semiconductor transport equations are solved together with the compact models in one circuit.

For the description of circuits the inheritance feature of the Input Deck is extensively used. The basic idea is that circuits consist of subcircuits which consist of subsubcircuits and so on. Each circuit is defined as a self-contained circuit and forms a black box. Black boxes and devices can again be used to form another circuit.

A circuit is described within a section. Its components may be subcircuits, devices, or compact devices. Each component is described by a section. The components of a circuit are usually inherited from the black boxes defined before. Variables inside a circuit definition form the contacts. There are two types of contacts: Inner contacts which are only used within the current circuit and outer contacts which form the input and output ports. To connect the components of a circuit, their outer contacts are assigned the names of the contacts of the circuit. The global circuit to be simulated is defined in the section called `Circuit`. This approach allows the definition of circuits of any complexity.

7 Stepping Functions

Stepping of keywords is also handled in a very flexible way. Nearly every keyword in the Input Deck database can be stepped by either providing minimum and maximum values, tables or lists of values, or values read from a data file. The latter feature is very useful when measured data has to be resimulated because the bias conditions can be directly fetched from the data file.

Stepping can be used to validate models as MINIMOS-NT allows physical models to be tested for a given set of parameters. Stepping these parameters enables to analyze models very effectively.

The stepped values are updated after each successful simulation of a given point. The stepping functions are implemented in MINIMOS-NT and made available in the Input Deck database as application specific functions. Thus, when the stepped values are changed user-defined feedback loops emerge.

8 Extern Section

In the so called `Extern` section several internal states of MINIMOS-NT are stored during simulation. Most of these states are parameters which show the progress of the solving procedure, e.g. the right hand side norm, the update norm, the CPU time, or the iteration counter. This information is needed to formulate the `while` and the `failure` condition for the iteration blocks which allows a user-defined feedback loop.

9 Iteration Schemes

Due to the complex interaction between the coupled partial differential equations the idea of iteration schemes is relatively old. Instead of solving the fully coupled equation system by a Newton method, the equations are solved consecutively in suitable blocks until convergence is reached. A classic example is Gummel's algorithm [9], for other examples see e.g. [10, 11, 12]. Each of these schemes has its advantages under specific circumstances, e.g. the bias conditions and the type of the device. As it is impossible to implement all useful schemes, a different solution has been sought in MINIMOS-NT which makes heavy use of its powerful input language. Iteration schemes are defined to consist of iteration blocks. These blocks can be arbitrarily nested. For each block the user can specify:

- Which equation assembly models to use: This offers a flexible way to create suitable equation sets.
- Which quantities of the equation set should be used as solution variables.
- A convergence criterion: This can be formulated using information about the full or partial update norms, the residuum of the complete equation set or of a suitable subset of equations, the iteration counters, ore other parameters.
- A failure criterion: This is needed to trigger timestep and stepping control.
- An enter criterion: This allows to use the block under certain circumstances only, e.g. for the first time step or when the norm of the last iteration block was smaller than a certain value.
- The damping scheme to use: Depending on the equation type or the bias condition a different damping scheme or a different parameter set might be advantageous.

The iteration blocks are processed in order. Before each iteration the convergence condition which is explained above called `while`-condition and the failure criterion of the active block are tested. If the failure condition evaluates to true the execution is terminated with an error state. If not and the convergence condition evaluates to true another iteration will be performed otherwise the iteration block is terminated successfully and the next block is processed. The conditions can be arbitrary expressions. Normally the runtime information the simulator writes back to the database is used (see Section 9) to formulate the conditions. An iteration block does not need to specify an equation set in which case they are used as grouping constructs. With these features an individual iteration scheme can be tailored to solve even such complex problems as electro-thermal mixed-mode simulations [13].

10 Interactive Mode

In the past two years MINIMOS-NT has been extended to a fully multi-dimensional device simulator which is able to simulate arbitrarily shaped two- and three-dimensional structures. Due to the increased computational costs which occur especially in three-dimensional simulations it becomes necessary to control the simulation flow during simulation.

Therefore, MINIMOS-NT now provides an interactive mode which can be entered by pressing the control-sequence `control-C` during simulation. The user is prompted a menu and may change entries in the database directly using simple commands similar to Unix shell commands, e.g., `ls` to list the contents of a section or `set` to set a variable. This new powerful mode of MINIMOS-NT enables, for instance, to change the parameters of the iteration scheme, to choose a different one, or to change the stepping behavior during runtime.

11 Conclusion

The demands on a control system for state-of-the-art TCAD applications like MINIMOS-NT increase with the complexity of the simulator and with the complexity of the user interface. A new kind of an object-oriented dynamic database, the Input Deck database, has been developed which is used to control the simulator, e.g., various simulation modes, the complete simulation flow, and the applied models.

The Input Deck database provides an inheritance mechanism which is effectively used to support a powerful default system, the definition of a hierarchical material database and the description of complex circuits. MINIMOS-NT uses abstract materials which enables the user to add new materials in an easy way. Since the Input Deck database stores expressions read from the description files rather than evaluating them at parse time and since evaluation takes place when keywords are queried feedback loops are possible as the simulator writes back internal states to the database during runtime. These feedback loops are used, e.g., to define well adjusted iteration schemes. Application specific functions registered in the database enlarge the functionality of its description language and are used to control stepping and to access values of stepped states of the simulator. MINIMOS-NT supports an interactive mode which allows full control over nearly all modules and each part of the simulation flow. Thus, the control system of MINIMOS-NT is very powerful and makes it possible to obtain highest possible flexibility while a consistent user interface is guaranteed.

References:

[1] IμE, *MINIMOS-NT 2.0 User's Guide*. Institut für Mikroelektronik, Technische Universität Wien, Austria, 2002.

[2] R. Klima, T. Grasser, T. Binder, and S. Selberherr, Controlling TCAD Applications with a Dynamic Database, *Software Engineering and Applications*, 2000, pp. 103–112.

[3] R. Klima, T. Grasser, and S. Selberherr, Controlling TCAD Applications with an Object-Oriented Dynamic Database, *15th European Simulation Multiconference*, 2001, pp. 161–165.

[4] T. Binder and S. Selberherr, Object-Oriented Wafer-State Services, *14th European Simulation Multiconference*, Society for Computer Simulation International, 2000, pp. 360–364.

[5] T. Binder and S. Selberherr, Object-Oriented Design Patterns for Process Flow Simulations, *Software Engineering and Applications*, 2000, pp. 159–166.

[6] H. Ceric, T. Binder, A. Hoessinger, C. Hollauer, and S. Selberherr, FEDOS - Rigorous Object Oriented Three-Dimensional Process Simulator. To be published in First International SiGe Technology and Device Meeting, 2003.

[7] R. Mlekus and S. Selberherr, Object-Oriented Algorithm and Model Management, *Intl. Conf. on Applied Modelling and Simulation*, 1998, pp. 437–441.

[8] T. Grasser, *Mixed-Mode Device Simulation*. Dissertation, Technische Universität Wien, 1999. http://www.iue.tuwien.ac.at/.

[9] H. Gummel, A Self-Consistent Iterative Scheme for One-Dimensional Steady State Transistor Calculations, *IEEE Trans.Electron Devices*, Vol. ED-11, 1964, pp. 455–465.

[10] W. Engl and H. Dirks, Numerical Device Simulation Guided by Physical Approaches, *Numerical Analysis of Semiconductor Devices and Integrated Circuits*, Vol. I, Boole Press, 1979, pp. 65–93.

[11] B. Meinerzhagen, K. Bach, I. Bork, and W. Engl, A New Highly Efficient Nonlinear Relaxation Scheme for Hydrodynamic MOS Simulations, *Workshop on Numerical Modeling of Processes and Devices for Integrated Circuits*, 1992, pp. 91–96.

[12] T. Simlinger, R. Deutschmann, C. Fischer, H. Kosina, and S. Selberherr, Two-Dimensional Hydrodynamic Simulation of High Electron Mobility Transistors Using a Block Iterative Scheme in Combination with Full Newton Method, *Fourth Intl. Conf. on Solid-State and Integrated-Circuit Technology*, 1995, pp. 589–591.

[13] T. Grasser and S. Selberherr, Fully-Coupled Electro-Thermal Mixed-Mode Device Simulation of SiGe HBT Circuits, *IEEE Trans.Electron Devices*, Vol. 48, No. 7, 2001, pp. 1421–1427.