*SIMULATION OF SEMICONDUCTOR DEVICES AND PROCESSES Vol. 5*
Edited by S. Selberherr, H. Stippel, E. Strasser – September 1993


# Mixed-Mode Multi-Dimensional Device and Circuit Simulation

J. Litsios, S. Müller, and W. Fichtner

Integrated Systems Laboratory, ETH-Zürich
Gloriastraße 35, CH-8092 Zürich, SWITZERLAND

Abstract

A mixed-mode device and circuit modeling environment has been developed that allows the combined simulation 1D, 2D and 3D devices together with SPICE-like circuit elements. The work is based on the device simulator SIMUL [1] as well as the circuit simulator CAzM [2]. By extending SIMUL to handle simultaneously more than one device as well as giving it access to the circuit simulator's functionalities it has been possible to combine both simulators' features into a single program (1D, 2D, 3D devices, extensive physical models for power devices, thermal-electric effects and all basic SPICE models).

## 1. Introduction

Mixed-mode device and circuit simulators (MDC simulators) respond to the need of both device and system designers in the sense that device designers want to know how devices respond when surrounded by circuits and system designers want to design circuits in which some devices have no usable analytical model. To satisfy this large spectrum of users, MDC simulators need to offer as many features as the individual device and circuit simulators. This creates an ever increasing complexity on the part of the MDC simulators.

We describe here an MDC simulator that allows mixing 1D, 2D and 3D devices as well as circuit-level elements. The work is based on the device simulator SIMUL [1] as well as the circuit simulator CAzM [2]. While other mixed-mode device and circuit simulators have been presented ( MEDUSA [3], GIGA [4], GENSIM [5], PISCES [6], CODECS [7]), the originality of this work lies specifically in the possibility to simulate together many devices of different dimensions as well as in its support of a full circuit simulator like CAzM.

Previous designs of MDC simulators have focused on connecting existing device and circuit simulators in a fairly rigid fashion. One of the main goals in the development of SIMUL has been flexibility. This has resulted in a very modular program that tries to separate concepts such as mesh dimension, physics, non-linear algorithms, linear solvers or sparse matrix formats.

The next section gives an overview of the organization of the simulator, followed by a description of the program's functionalities and input language. Finally some examples are presented.

## 2. Organization of SIMUL

A simplified structure of the simulator can be seen in Fig. 1. During a Newton-type solution the device and circuit equations are assembled together to form a linear system's right-hand side and Jacobian. SIMUL is written in C++ and uses an object oriented design. New equations are introduced as objects and are linked together through a data manager that works out their dependencies. This object oriented design of the code assures a large independence in the different parts of the code. Thus, for example, the non-linear solver does not need to know which linear solver it is using or the physical models do not need to know the dimension of the underlying mesh. It is this design philosophy which gives SIMUL its generality allowing, for example, the program to be linked simultaneously to many circuit simulators. Currently SIMUL is connected to the CAzM simulator and to a yet unnamed circuit simulator developed in our laboratory.

A key concept used in SIMUL is the possibility to merge variables from the different equations. This technique is inspired from the node merging used in the modified node analysis (MNA) method for circuit analysis. We define the merging of two variables x and y as replacing every occurrence of y by x and replacing the two equations associated with x and y by a single equation. Usually these equations represent the current, heat flow or charge mismatch, thus the merged equation is the sum of the individual equations. The use of variable merging allows the different device and circuit elements to be connected through the merging of connectivity nodes. Figure 2 shows how this is used to connect two devices with a resistive or non-resistive contact. Variable merging also facilitates implementing device structures that share common values such as floating gates.

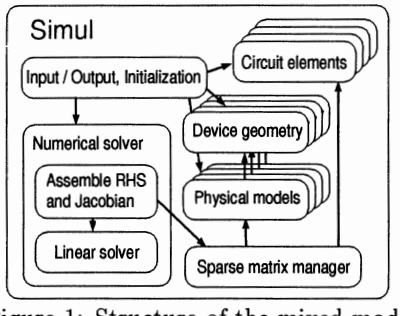



Figure 1: Structure of the mixed-mode multi-dimensional SIMUL

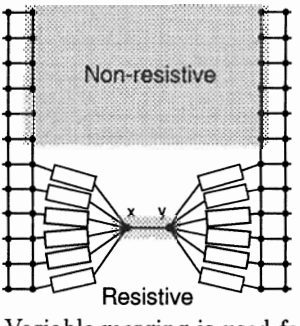



Figure 2: Variable merging is used for resistive and non-resistive device coupling

## 3. Functionality and Input language

The SIMUL input language offers all the necessary flexibility to perform complex simulations. Figure 3 shows an example of the SIMUL input language. The different device parameters such as the selection of the physical models or the desired precision can all be given independently for the different devices or circuits. A connectivity netlist is given as well as a series of numerical solve operations to be performed. The solver supports both coupled equations as well as Gummel iterations and a two level Newton solver (uses non-linear elimination). These can be performed on single devices, the full system or a specific set of devices and circuit elements. This flexibility is often necessary to construct a correct initial solution. Finally, SIMUL allows transient, quasi-stationary and small-signal analysis simulations using either a direct, iterative or domain decomposition linear solver. The domain decomposition

solver splits the linear solve over the devices and the circuit and allows pivoting in the circuit part of the system.

```
System {
    Device "mct" "v1" (1="anode")
    Device "diode" "dio" (0="high" 1="anode")
    CAZM ("anode"="anode" "high"="high"){
        ...
        "I1 anode 0 -0.25e-5"
        "v1 high 0 5000"
        ...
        }
    }
Solve {
    Coupled { "v1".Poisson "v1".Electron "v1".Hole Circuit }
    Coupled { "dio".Poisson "dio".Electron "dio".Hole }
    Coupled { Poisson Electron Hole Circuit }
    Transient ...
    }
```

```
Device "mct" {
    Electrode {
        { Number=0 Voltage=0 }
        ...
        }
    File {
        Grid="v1"
        ...
        }
    Physics {
        Recombination (
          SRH(DopingDependence)
          Auger Avalanche)
        ...
        }
    ...
    }
```

Figure 3: Example of SIMUL's input language

## 4. Examples

Three examples of simulation results are presented. Figure 4 demonstrates the mixed-mode multi-dimensional features of SIMUL with a transient simulation of a recitifier in which each diode is modeled differently with a 1D, 2D or 3D device or with a SPICE model. Figure 5 shows the schematic used in simulating the influence of the gate resistance (Rg) on an IGBT's turn-off behavior when connected to a diode and snubber circuitry. Here the IGBT and the diode are simulated as 2D devices (700 and 300 vertices), the snubber diodes are 1D devices (228 vertices) and the additional circuitry use circuit models. Two simulation were performed with a gate resistance of 2Ω and 20Ω. They required 3972s (97 time steps) and 3329s (91 time steps), respectively, on a SPARCstation 10.

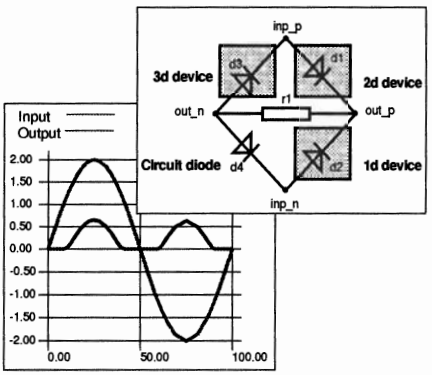



Figure 4: Simulation of a multi-dimensional mixed-mode rectifier

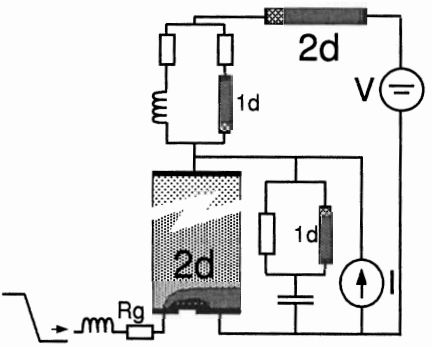
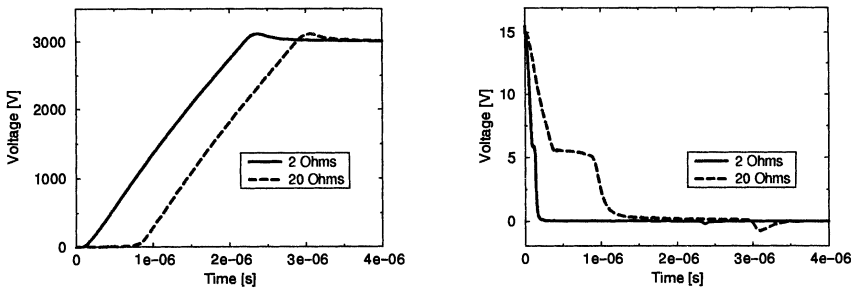



Figure 5: Schematic of the IGBT turn-off example

Figure 6a and 6b show the influence of the gate resistance on the IGBT's anode voltage and gate voltage when the gate is ramped from 15V to 0V in 500ns. Figure 7 shows the influence on the current through the IGBT.

Finally, Fig. 8 shows MOS C-V curves that were computed using SIMUL's small-signal capabilities. These curves were computed by connecting a MOS diode to a circuit

voltage source. A quasi-static ramping of the voltage source is performed in which a small-signal analysis is computed.




(a) (b)

Figure 6: Influence of the gate resistance on the IGBT's switching behavior: (a) on the anode voltage, (b) on the gate voltage.

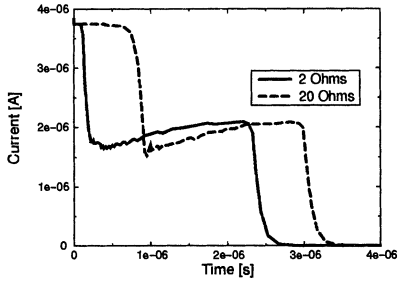



Figure 7: Influence of the gate resistance on current through the IBGT.

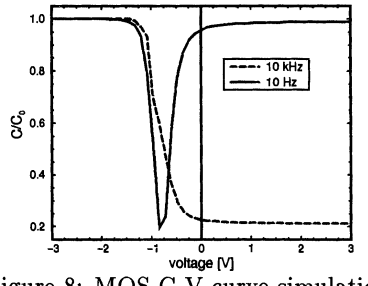



Figure 8: MOS C-V curve simulation with SIMUL.

## Acknowledgements

This work has been financially supported by the ESPRIT-6075 (DESSIS) Project.

## References

[1] IIS, *Simul Manual.* Integrated Systems Laboratory, ETH Zurich, Switzerland, 1.0 (alpha) ed., 1992.

[2] W. M. Coughran, Jr., E. Grosse, and D. J. Rose, "CAzM: A circuit analyzer with macromodelling," *IEEE Trans.*, vol. ED-30, pp. 1207–1213, 1983.

[3] Institut für Theoretische Elektrotechnik, Technical University Aachen, Germany, *MEDUSA User's Guide.*

[4] B. Freydin, E. Velmre, and A. Udal, "Giga - power semiconductor devices simulation software," *The Exterminator, Silvaco International,* 1990.

[5] J. Bürgler and W. Fichtner, *GENSIM User's Guide.* Integrated Systems Laboratory, ETH Zurich, Switzerland.

[6] M. R. Pinto, C. S. Rafferty, and R. W. Dutton, *PISCES II: Poisson and Continuity Equation Solver.* Stanford University, Stanford, CA 94305, 1984.

[7] K. Mayaram Tech. Rep. UCB/ERL M88/71, Electronic Research Laboratory, University of California at Berkeley, 1988.