

Software Tools for Three-dimensional Device Simulation

P. Conti G. Heiser M. Westermann W. Fichtner
Integrated Systems Laboratory, ETH Zürich, Switzerland¹

Silicon devices are inherently three-dimensional (3-d) structures. While for many problems the behavior of devices can be modeled in either one or two dimensions, for MOS and bipolar devices with submicron design rules, complex latchup structures, or DRAM cells, 3-d simulation becomes necessary.

Three-dimensional modeling is considerably more complex than the solution of the equivalent 2-d problem. Major difficulties include the allocation of spatial grids, the visualization of output data and the solution of the discretized equations, where conventional numerical methods turn out to be inadequate.

We have developed a software package for the simulation of complex semiconductor devices in 3-d. The package includes the grid generator Ω , the simulator `SECOND` and the interactive graphics program `PICASSO`. In the first part of the paper we describe the capabilities of each of these components. In the second part, we illustrate CPU and memory requirements for the simulation of a complex device with several tens of thousands of points.

The grid generator Ω allows to tessellate arbitrary plain faced structures. Meshes are generated according to a user-supplied point density function, either based on the doping distribution gradient or with explicit mesh sizes for subregions of the device. Ω has no privileged direction; the point density may vary arbitrarily in each direction, and the resulting mesh will show the required point densities while avoiding redundant points.

Ω is based on a modified octree approach. Bricks either considered too coarse according to the user-supplied density requirements, or crossing device boundaries, are iteratively refined until the required point density and a sufficient approximation of the boundary are achieved. Bricks are refined by subdivision in eight similar bricks. Then bricks lying between coarse and dense mesh regions, or across the device boundaries, are subdivided in tetrahedra, quadrilateral pyramids, prisms and bricks.

In addition to the grid itself, Ω provides the dual lattice needed for the box method integration of the device equations. The underlying algorithm avoids the well-known obtuse angle problem. All surfaces of the Voronoi diagram are perpendicular to edges of the generated grid; therefore, box segments perpendicular to the edges of the grid are positive.

The simulator `SECOND` solves the classical semiconductor equations of the drift-diffusion model. The discretized Poisson equation and current equations for both carriers can be solved using either the decoupled Gummel scheme or a fully coupled Newton scheme. In the latter case, a few Gummel iterations are performed first to obtain a reasonable initial solution for the coupled scheme. Both stationary and transient simulations can be performed.

The simulator incorporates the usual models for mobility degradation due to heavy doping effects, velocity saturation and normal field effects at gate oxide interfaces. Recombination effects due to localized traps and three particle (Auger) recombination effects are modeled.

The linearized equations are solved with preconditioned conjugate-gradients type methods. For time integration, a composite trapezoidal/backward differentiation formula is used.

`PICASSO` is an interactive program for the visualization of simulation results. Plots show the computed values on the surface of and on user-defined cuts through the device. The interactive capabilities of `PICASSO` include in particular the choice of a suitable view of the device through rotation, translation and scaling of the object, the selection of individual device regions (e.g. only the silicon part) for visualization and the choice of the range and scaling (linear or logarithmic) of the color map. A batch mode is provided for the computation of large numbers of pictures, e.g. for the generation of "movies" for transient runs.

Ω and `PICASSO` are both implemented in the object-oriented programming language C++. `PICASSO` is built on top of `InterViews`, an object-oriented graphics library developed for X-Windows at Stanford University. `SECOND` is implemented in `FORTRAN` and vectorizes and parallelizes well on a broad range of supercomputers. Great emphasis was put on the portability of the code. So far `SECOND` has been installed on Alliant, Convex, Cray, Fujitsu-VP200 and Sun computers. Ω and `PICASSO` have been ported to Alliant, Apollo, Sequent and Sun.

¹Address of the authors: Integrated Systems Laboratory, ETH-Zentrum, 8092 Zürich, Switzerland.
 Phone: +41 1 256 23 48. Fax: +41 1 252 09 94.

Three-dimensional simulations of complex devices are both very CPU- and memory-intensive. To illustrate CPU and memory requirements at each stage of the simulation process, we consider the oxide-isolated ECL bipolar transistor shown in Fig. 1. In a first run, we have applied the voltages $U_{BE} = 0.6V$ and $U_{CE} = 5V$. In a second run, we have stepped U_{BE} from 0.6V to 0.8V. Figure 1 shows the doping distribution and current density for the bias of 0.8V.

The grid for the above example consists of 96k mesh points, 60k bricks and 120k tetrahedra, pyramids, and prisms. It was generated on a Sun SPARCSTATION 1 with 16 MB RAM in 10 minutes and required 20 MB of virtual memory.

The proper simulations were performed on a mini-supercomputer Convex C-220 with 256 MB RAM, using one processor. From our experience, this machine is about 10 times slower than a Cray-2. The first run was done using a decoupled Gummel scheme; the second run was done with a coupled Newton method. Table 1 summarizes the run times and memory requirements for the two runs. Note that the two runs should not be compared against each other, since the applied biases are different.

PICASSO requires 80 MB of virtual space for the visualization of the above example. Since it is based on X-windows, it can be run remotely on a big machine while the images are plotted on a workstation. Loading the grid and the simulation results and building internal data structures takes about 5 minutes on a Sun 4/280; the proper plotting of a data set takes about 30 seconds.

The above example shows that 3-d steady-state simulations of complex silicon devices with 100k points meshes can be done within minutes on a supercomputer or hours on a mini-supercomputer. Both grid allocation and visualization of simulation results can be performed in reasonable time on modern workstations. However, really interactive pre- and postprocessing with virtually no waiting times would also require supercomputer capabilities.

	tot. time	virtual memory	# Gummel loops	# Newton iterations
Gummel	90 min.	244 MB	7	—
Coupled	160 min.	490 MB	2	5

Table 1: Run times (Convex C-220) and memory requirements for one working point of the bipolar transistor shown in Fig. 1

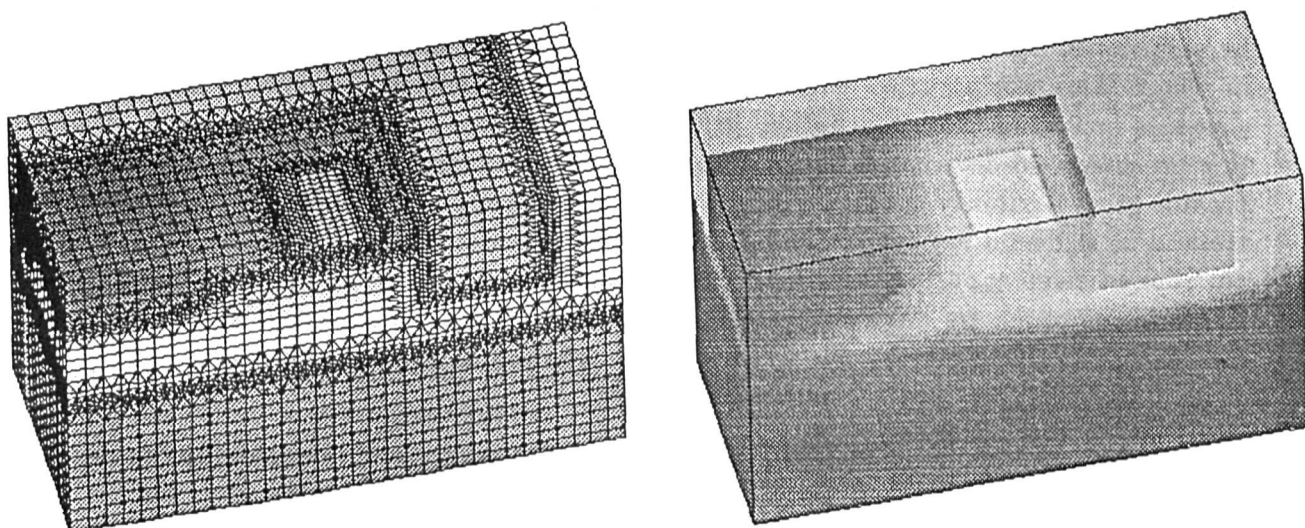


Figure 1: Doping profile and electron current in ECL bipolar transistor