Extended Abstracts of the 22nd (1990 International) Conference on Solid State Devices and Materials, Sendai, 1990, pp. 143-146

C-1-4

Three Dimensional Transient Simulation of Complex Silicon Devices

Paolo Conti Gernot Heiser Wolfgang Fichtner

Integrated Systems Laboratory, ETH Zürich, 8092 Zürich, Switzerland

Abstract

We present a general-purpose program for the simulation of semiconductor devices in three dimensions. This program solves the Poisson and continuity equations in steady-state and transient conditions. The implemented grid allocation scheme allows for spatial grid adaption in all directions. The linear systems are solved using preconditioned conjugate gradient-like methods. We describe the investigation of parasitic latchup in a CMOS structure and of the turn-off of a bipolar transistor with our program. Transient simulations on meshes with several tens of thousands of points can be performed within hours on a supercomputer.

1 Introduction

The rapid developments in the integrated circuit and optical communications industry are increasingly relying on the results of numerical simulations for faster and better prototyping. The numerical simulation of semiconductor devices has therefore become an important area of research and development in both academic and industry environments.

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, 3-d simulation becomes necessary for MOS and bipolar devices with submicron design rules,. complex latchup structures or DRAM cells. While for 2-d device simulation quite general programs exist by now, present 3-d device simulators [2, 3] are limited in many respects. The main problems include the modeling of general geometries, the allocation of spatial grids, the solution of the linearized equations with efficient and stable iterative methods and the visualization of simulation results.

This paper describes our approach towards the simulation of realistic 3-d semiconductor devices. Section 2 formulates the well-known device equations. In section 3 we describe how we discretize the 3-d domain using tetrahedra, pyramids, prisms and bricks. Section 4 shows how the equations are linearized and how the linear equations are solved using iterative conjugate gradient-like methods. Finally, section 5 shows results from latchup investigations in a CMOS structure and from the simulation of the turning off of a bipolar transistor.

2 Semiconductor Equations

Our 3-d device simulator SECOND solves the conventional semiconductor drift-diffusion equations:

$$\nabla \cdot (\epsilon \nabla \psi) + q(p - n + N) = 0 \, , \tag{1}$$

$$\frac{1}{q}\nabla \cdot \mathbf{J}_n - R_n = \frac{\partial n}{\partial t} \, , \tag{2}$$

$$-\frac{1}{q}\nabla \cdot \mathbf{J}_p - R_p = \frac{\partial p}{\partial t} \, , \tag{3}$$

where the dependent variables to be determined are the electrostatic potential, $\psi(\mathbf{x})$, and the electron and hole carrier concentrations, $n(\mathbf{x})$ and $p(\mathbf{x})$, respectively. Here, q, ϵ, N, and R are electron charge, dielectric constant, net impurity (doping) concentration, and recombination-generation terms, respectively. The variables $\mathbf{x}$ and t stand for the space and time variables. The electron and hole current densities are given by

$$\mathbf{J}_n = -q\mu_n n \nabla \psi + k_B T \mu_n \nabla n, \tag{4}$$

$$\mathbf{J}_p = -q\mu_p p \nabla \psi - k_B T \mu_p \nabla p \tag{5}$$

with the mobilities μ, the temperature T and the Boltzmann constant k_B.

We employ the usual models incorporating velocity saturation and heavy doping effects, as well as normal field effects at the gate oxide interface in MOSFET inversion layers. The recombination terms include recombination at localized traps and three-particle (Auger) effects.

3 Discretization

The device equations are spatially discretized by the box method (BM). We use grids consisting of tetrahedra, quadrilateral pyramids, prisms and bricks, as created by our grid generator Ω [4]. Ω is based on a modified Octree technique; to generate a grid it first tessellates the device with the required density using bricks of the appropriate size. The other element types are then used to pass from coarse to dense mesh regions without hanging nodes. Pyramids and prisms are also used to fit non-rectangular device boundaries and material interfaces. This results in smoothly varying grid densities, avoids small angles that could lead to numerical problems and produces regular grids in device regions where a constant point density suffices. The density of the generated grid depends on the doping distribution and on user requirements.

In addition to the spatial grid, Ω provides the dual grid (Voronoi diagram) required for the box discretization. The box surfaces are guaranteed to be positive by construction of our Octree-based grids. Therefore, we never encounter the well-known obtuse angle problem inherent in the BM.

For the time discretization we use the scheme proposed by Bank et al. [1] which uses a composite trapezoidal/backward differentiation formula for the time integration and a time step control based on an estimate of the local truncation error.

4 Numerical Aspects

For the linearization of the non-linear equations resulting from the spatial discretization we either use the (decoupled) Gummel iteration, where repeatedly each PDE is solved individually until a self-consistent solution is obtained, or a Newton iteration on the full (coupled) system of equations. In the Gummel case the individual non-linear systems are also linearized by a Newton method. The Gummel method has the advantage that linear systems with only N unknowns must be solved, where N is the number of grid points, while the coupled Newton iteration requires the solution of $3N$-dimensional linear systems. This means that the memory requirements are lower for the Gummel method, and it also tends to be faster for quickly converging problems. On the other hand it converges very slowly, or not at all, in cases of strong coupling between equations. The full Newton scheme, on the other hand will only converge if started sufficiently close to the solution.

We therefore use the Gummel method for problems with low current flow, as in DRAM simulations, and to obtain a suitable starting point for the full Newton scheme. For transient simulations or for devices in a high current mode the full Newton scheme must be used.

For the Gummel iterations it is advantageous to use the electric potential ψ and the carrier densities n and p as the unknowns, since then the continuity equations are almost linear and hence converge quickly. However, in the coupled case this choice of variables is not appropriate due to the fact that the densities vary by many orders of magnitude stronger than the electric potential. We therefore replace the densities by the quasi-Fermi levels.

Realistic 3-d simulations typically require grids with several tens of thousands to more than a hundred thousand grid points. The resulting linear systems are far too big for direct (Gaussian elimination based) solvers and iterative methods must be used. Since the linear systems arising from the current continuity equations and from the fully coupled Newton iteration are usually ill-conditioned, most popular iterative solution methods do not work.

We have found that the conjugate gradient squared method (CGS), together with ILU preconditioning, usually performs best. However, even this method tends to fail to converge due to truncation problems, unless the implementation is done carefully [5].

The irregularity of our grids makes it more difficult to get good performance out of vector and parallel computers. Reordering of the unknowns is required so that independent equations can be processed in parallel. This has an adverse effect on the condition of the preconditioned system and results in increased iteration counts. However, we found that in general these losses are more than offset by the resulting gain in machine performance.

5 Results

We present two examples of transient simulations performed with the programs Ω and SECOND.

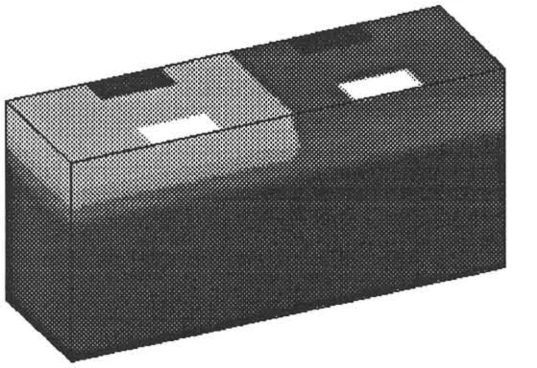

Figure 1: Impurity distribution for the CMOS structure

The first one is an investigation of parasitic latchup. We have examined part of an n-well CMOS configuration in 1 μm technology. The example features an extremely shallow n-well (1.35μm). Figure 1 shows the doping distribution and the rectangular arrangement of the various tubs.

The structure contains two parasitic bipolar elements: a lateral npn and a vertical pnp transistor. If the device did latch, the thyristor current would flow diagonally from the p^+ to the n^+ diffusion. In the simulation we applied 5 *V* to the n-well contacts and 0 *V* to the p-well contacts and the substrate. We then turned on the lateral transistor by applying a voltage pulse to the p^+ diffusion, rising from 0 *V* to −0.85 *V* in 1 *ns*. Figure 2 shows the electron current density after the pulse has been held for 10 *ns* — there is no indication of latchup.

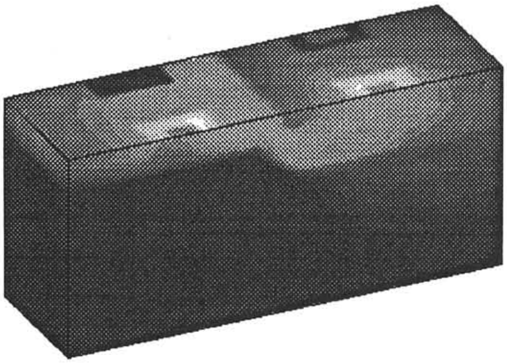

Figure 2: Electron current density after 10 *ns*

A second example simulates the turning off a bipolar transistor. Figure 3 shows doping and geometry of a trench-isolated npn ECL transistor. The grid used for the simulation is shown in Fig. 4. We used a common base configuration with a collector base bias of 5 *V* and an emitter base bias of 0.8 *V*. The transistor was switched off by linearly reducing the emitter bias to 0 *V* within 1 *ns*. Figures 5 and 6 show the electron current density in the device at different times. The figures indicate that the transistor is basically turned off after as little as 0.2 *ns*.

The simulations were performed on a Convex C-220 with 256 M bytes of memory and on a Cray-2 with 1 G byte, using one processor on both machines. Table 1 summarizes the time and memory requirements and gives some figures on the convergence behavior of both simulations. In general we find the Cray-2 about 8 – 10 times faster than the Convex.

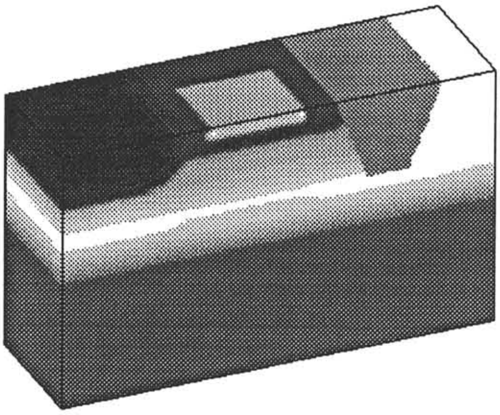

Figure 3: Impurity distribution of the bipolar transistor

Structure	CMOS	Bipolar
machine	C-220	Cray-2
number of grid points	34673	31592
memory *[M byte]*	128	160
total CPU time *[h]*	25	2
time steps	168	83
average time step *[ps]*	65	13
Newton iterations per time step (average)	2.0	2.6
average number of linear iterations	25	62

Table 1: Time and memory requirements and convergence behavior for CMOS structures

6 Conclusions

We have presented our approach for solving the drift-diffusion equations in 3-d. The implemented grid allocation strategy allows both efficient allocation of grid points and the modeling of general device geometries. We have discussed our experiences with sparse iterative solvers and the impact of preconditioning on the convergence behavior. We have used our simulator for the investigation of parasitic latchup in CMOS structures. The example shows that transient simulations on grids with 50k mesh points lie on the edge of todays mini-supercomputers.

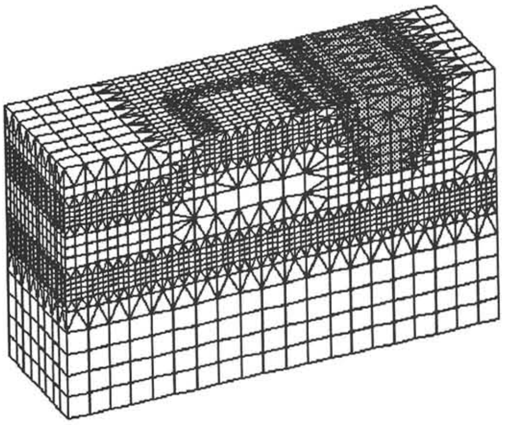

Figure 4: Simulation grid for the bipolar transistor

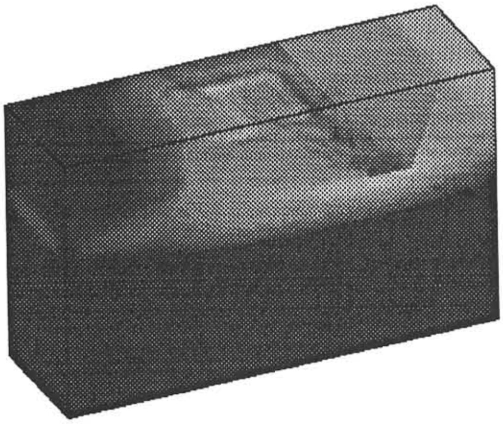

Figure 5: Electron current density after 0.1 *ns*

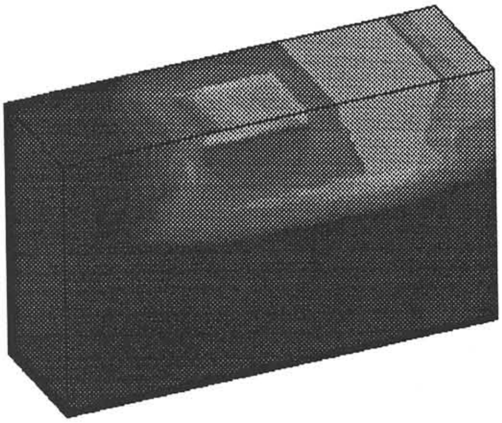

Figure 6: Electron current density after 0.2 *ns*

References

[1] R. E. Bank, W. M. Coughran, Jr., W. Fichtner, E. H. Grosse, D. J. Rose, and R. K. Smith. Transient simulation of silicon devices and circuits. *IEEE Trans. CAD/ICAS*, CAD-4:436–451, 1985.

[2] E. M. Buturla, P. E. Cottrell, B. M. Grossman, and A. K. Salsburg. Finite-element analysis of semiconductor devices: The FIELDAY program. *IBM J. Res. Develop.*, 25:218–239, 1981.

[3] J.-H. Chern, J. T. Maeda, L. A. Arledge, and P. Yang. SIERRA: A 3d device simulator for reliability modeling. *IEEE Trans. on CAD/ICAS*, 8:516–527, 1989.

[4] P. Conti, N. Hitschfeld, and W. Fichtner. Ω – an octree-based mixed element grid allocator for adaptive 3d device simulation. Submitted to IEEE Trans. on CAD/ICAS.

[5] G. Heiser, C. Pommerell, J. Weis, and W. Fichtner. Three-dimensional numerical semiconductor device simulation: Algorithms, architectures, results. Submitted to IEEE Trans. on CAD/ICAS.