Parallel 3D Finite Element Power Semiconductor Device Simulator Based on Topologically Rectangular Grid

A.R. Brown, A. Asenov, S. Roy and J.R. Barker

Department of Electronics and Electrical Engineering
Glasgow University
Glasgow, G12 8QQ, Scotland, UK

Abstract

Here we report on the development of a new parallel, scalable and portable 3D finite element power semiconductor device simulator. The emphasis in the design of this simulator is placed on the FE grid generation, on the optimised parallel generation and assembly of the discretization matrices, and on the development of a suitable, scalable linear solvers. For discretization use topologically rectangular FE grid based on non-rectangular bricks.

1. Introduction

The cellular structure of most power devices requires a 3D solution of the basic semiconductor equations. The octagonal or hexagonal shape of typical power MOSFET, IGBT or MCT cells [1] and their complex doping distributions require a finite element (FE) discretization. In many cases more than one cell should be included in the simulations in order to obtain an adequate description of the device behaviour. The size and the computational complexity of the problem make it a distinguished candidate for massively parallel implementation. Only recently has a more systematic approach been applied to the design of parallel device simulation codes [2]. To achieve better results the design of the parallel simulation software should reflect the architecture of the parallel platforms.

Here we report on the development of a parallel, scalable and portable 3D finite element power semiconductor device simulator. It is based on a spatial decomposition of the simulation domain over an array of processors [3]. This approach minimises the interprocessor communications by reducing the ratio between the bulk and the surface of the partition subdomains. The emphasis is placed on the generation of topologically rectangular FE grids amenable to the domain decomposition approach, on the optimised parallel generation and assembly of the discretization matrices, and on the development of suitable, scalable linear solvers.

2. Spatial Device Decomposition

Our parallel power semiconductor device simulator is designed for Multiple Instructions Multiple Data (MIMD) parallel computers with distributed memory. It is based on the spatial device decomposition approach .

To enhance the portability of the simulator, the code is split into two distinct parts: a hardware dependent communications harness and the simulation engine. The communications harness provides all global and local communications between the processors necessary for the proper operation of the simulation engine. The simulation engine is designed to operate on an arbitrary 1D, 2D or 3D array of processors including a single processor. This means that the solvers are virtually independent of the processors topology if the necessary global and local communications are provided.

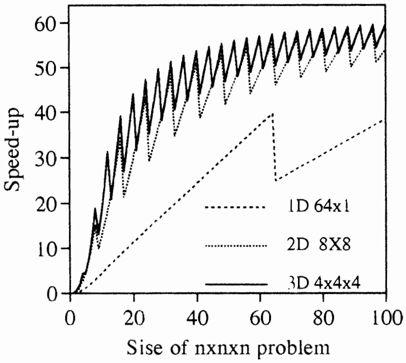

Figure 1: Theoretical speed-up of a hypothetical 3D linear solver based on a 1D, 2D and 3D array of processors

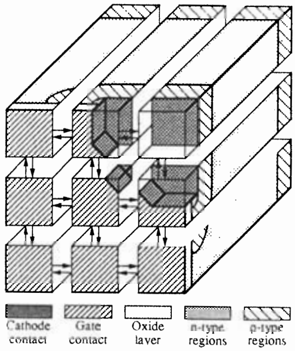

Figure 2: Partition of the 3D device simulation domain on a 2D array of mesh connected processors

The simulator can work with both a rectangular finite difference grid and a topologically rectangular finite element grid. This simplifies significantly the partitioning of the solution domain on the array of processors and the design of the communications harness. It is clear that the best processor configuration for spatial device decomposition of the topologically rectangular 3D grid is a 3D array of mesh connected processors. This is illustrate in Fig 1 where the theoretical speed-up of a hypothetical linear solver [4] is plotted as a function of the size of a *nxnxn* rectangular grid. The grid is partitioned on 64 processors organised in three different configurations: a 64 processor pipeline, an 8x8 2D array and a 4x4x4 3D array of processors. However we are restricted to a 2D array of processors on our Parsytec parallel computers. The spatial device decomposition of a 3D device on a 2D array of processors is illustrated in Fig 2. The solution domain is automatically decomposed into *N*x*M* subdomains in one grid plane (*i,j*). All corresponding grid point in the third grid direction *k* lie on the same processor. To achieve better speed-up each subdomain in the (*i,j*) plane should be as square as possible.

3. Solution Domain and Grid Generation

The solution domain and the generated grid depend on the structure of the simulated device and the doping concentrations inside. The grid generation in the solution domain proceeds on a single processor. After the grid generation the doping profile is assigned to the grid points, the material type is assigned to each finite element and the boundary conditions are identified. The generated grid with the doping, material and boundary conditions information is than distributed over the processor array. The distributed semiconductor solver is universal and independent of the particular device structure.

The grid generation approach is illustrated on an example a typical octagonal IGBT. Because of the symmetry the simulation domain involves only one quarter of the cell.

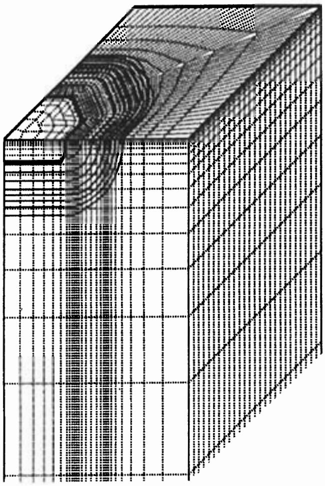

Figure 3: Solution domain and discretisation of octagonal IGBT. The boron profile in the cathode region is indicate

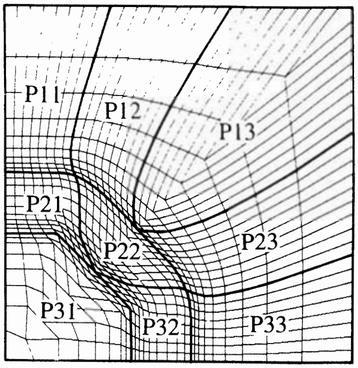

Figure 4: Partition of the (i,j) plane of the domain from Figure 3 on an array of 3x3 processors

The finite element grid is a topologically rectangular grid. It keeps the number of grid points along the grid lines constant in each one of the index directions *i* and *j*. The grid is based on distorted bricks. The grid generation process is determined by specified contours in the solution domain. In this particular example the guiding contours are the shape of the gate electrode and the metallurgical p-n junctions in the device. An example of the partitioning of the (*i,j*) plane of the solution domain on an array of 3x3 processors is given in Figure 4.

4. Parallel Discretization and Solution

We have adopted the decoupled Gummed procedure for the solution of steady-state problems and a modification of the decoupled Mock procedure based on time dependent version of the Poisson equation. The both schemes are simple and amenable to parallelization.

The Galerkin finite element approach has been adopted to solve the Poisson equation on a finite element grid. The integration over the distorted brick finite elements during the discretization was carried out by a linear isoparametric mapping of each element into an unit cube. For the parallel matrix generation and assembly we use a node based partition of the grid and node based assembly approach in which the solution subdomain on each processor is scanned not element by element, but node by node. This leads to almost 100% efficiency when the number of nodes in the *i* and *j* directions are divisible by the corresponding numbers of processors [4].To solve the nonlinear system arising from the discretization of the Poisson equation we have adopted a Block Newton SOR scheme. The parallel performance of the method is illustrated in Figure 5.

The discretization of the current continuity equation on the distorted brick finite element grid is more complicated. We are examining three possible approaches. The simplest one is to divide each distorted brick into six tetrahedral elements and to carry out a standard 3D control volume Gummel type of discretization. The second approach is to use a 3D analogue of the 2D Gummel like discretization developed for quadrilateral finite elements. Finally shape functions exponentially fitted to the potential distribution could be used. A parallel implementation of the BiCGSTAB(2) method has been adopted for the current continuity equation.

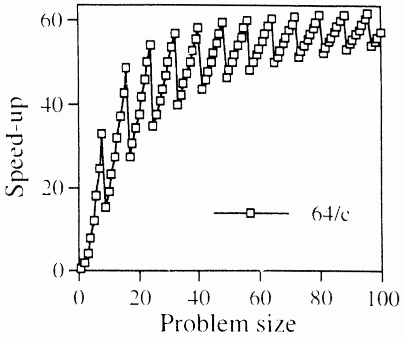

Figure 5: Speed-up of the Block Newton SOR method for a cubic *nxnxn* problem on an 8x8 array of transputers

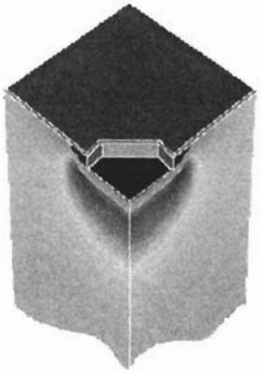

Figure 6: Electric field distribution in aoctagonal cell IGBT at 600V anode voltage

Finally an example of the electrical field distribution in a cellular IGBT at 600V anode voltage is illustrates in Fgure 6.

5. Conclusions

In this work we have presented our systematic approach to the design of a parallel finite element 3D power semiconductor device simulator. Our attempt was to build a portable and scalable parallel code which runs with high efficiency on variety of parallel platforms. To achieve this goal special measures were undertaken at each stage of the software development.

Acknowledgements

This work is funded by the EPSRC under grant GR/H23085 as part of a LINK/PEDDS project. We would like to thank Peter Waind and David Crees of GEC Plessey Semiconductors for their information and helpful discussions.

References

1. Brown, A R, Asenov A, Barker, J R, Jones, S, Waind, P, "Numerical simulation of IGBTs at elevated temperatures", *Proc. Int. Workshop on Computational Electronics*, Ed C. M. Snowden, University of Leeds Press, pp. 50-55, 1993.
2 T.F. Pana, E.L Zapata and D.J. Evans, "Finite element simulation of semiconductor devices on multiprocessor computers" *Parallel Computing*, vol. 20, pp. 1130-1159, 1993.
3. Asenov, A, Barker, J R, Brown, A R, Lee, G L: "Scalable parallel 3D finite element nonlinear Poisson solver", *Massively Parallel Processing Applications and Development,* Eds. L. Dekker, W. Smit and J.C. Zuidervaart, Elsevier, pp 665-672, 1994.
4. 7. Asenov A, Reid, D, Barker, J R: "Speed-up of scalable iterative linear solvers implemented on an array of transputers", *Parallel Computing*,Vol 20, pp 375-387, 1994.