SIMULATION OF SEMICONDUCTOR DEVICES AND PROCESSES Vol. 4
Edited by W. Fichtner, D. Aemmer - Zurich (Switzerland) September 12-14, 1991 - Hartung-Gorre

Spatial Domain Decomposition in Silicon Devices and Its Application To Transient Device Analysis

Howard Read and Andrzej Strojwas
Electrical and Computer Engineering Department
Carnegie Mellon University, Pittsburgh, PA 15232, USA

Shigetaka Kumashiro
VLSICAD Engineering Division, NEC Corporation, Japan

Abstract

The transient simulation of multiple semiconductor devices is critical in the analysis of dynamic effects such as latch-up. During transient simulation, a large system of coupled nonlinear partial differential equations (PDE) must be solved with considerable effort. In this paper, we present a method of decomposing the spatial domain of silicon (containing one or more devices) into smaller subdomains. A transient simulation algorithm can then use different time-steps within each subdomain and avoid a full solution of the nonlinear PDE's in areas where the solution does not change rapidly.

1 Introduction

The transient simulation of multiple semiconductor devices is critical in the analysis of dynamic effects such as latch-up and turn-on and turn-off transients [1] [2]. During transient simulation, a large system of coupled nonlinear partial differential equations (PDE) must be solved with considerable effort. A popular technique to deal with such large problems is a strategy called domain decomposition. This technique has been used successfully to solve many classes of problems including parabolic, hyperbolic and elliptic PDE's by dividing the domain of the problem into smaller subdomains which are solved separately and with considerably less expense [3], [4], [5].

In this paper, we present a method of decomposing the spatial domain of silicon (containing one or more devices) into subdomains. After decomposing the domain, the transient analysis algorithm uses different time-steps within each subdomain and avoids a full solution of the nonlinear PDE's in areas where the solution does not change rapidly. Although domain decomposition is often used with parallel algorithms, we show how it can be used without parallelization due to the relation between spatial and time domain *latencies*.

Section 2 of this paper describes some inherent difficulties in transient simulation and explores the concept of latency in time and space. Section 3 explains the concept of decomposition and gives a flexible algorithm to decompose a silicon domain based on charge transport considerations. In Section 4, we show the form of the equations after decomposition. In the last section, we show the results of applying the decomposition algorithm to a simulated MOS transistor.

2 Transient Simulation and Latency

The nonlinear PDE's which arise in transient semiconductor simulation form a stiff system of equations with widely varying time constants which are dependent on the dielectric relaxation time, τ_d :

$$\tau_d = \frac{\epsilon}{q\mu_n n + q\mu_p p} \tag{1}$$

Because the rate of spatial charge dispersal is exponentially proportional to τ_d we can expect regions of high charge density to react more quickly than those of low charge density, and in these regions a smaller time-step in the transient simulation will be necessary. Time Domain Latency (TDL) means that while some areas of the silicon domain are *active*, i.e., the rate of charge transfer is changing, other subdomains are *dormant* and the rate of charge transfer there is not changing as quickly. If we are able to separate these regions *a priori* then we can assign different time-steps to each.

Although TDL has been explored in other classes of problems, such as network theory [6] [7] [8], it is not as simple to exploit in device analysis because the boundaries between silicon subdomains are not as clear as they are in hierarchical or modular circuits. It may not be correct to divide a multi-device circuit along the boundaries of each device because this ignores the underlying concept of dielectric relaxation and charge transfer. Instead, we can use the work of Ohtsuki [9] where the domain is approximated by lumped-elements and with these approximations formulate decomposition criteria.

3 Motivation and Decomposition Principle

Consider the case in Figure 1. We first discretize the basic semiconductor equations using finite differences on a tensor product grid. Between each pair of nodes on the grid there exists an edge, b_b , with associated electron and hole currents, $J_n|_b$, $J_p|_b$, and potential ψ_b . Each mesh edge also has associated length and cross section, L_b and X_b .

As stated in [10], one common goal of decomposition is finding a useful geometric partitioning to exploit locality. The locality condition used in a semiconductor is charge transfer. We will cluster regions in the silicon where charge transfer occurs and make them the centers or *nuclei* of our subdomains. Between adjacent nuclei there will be regions of low charge transfer which in a circuit sense are equivalent to high impedances (or low admittances). By clustering the edges, b_b , of high admittance, we form the nuclei of subdomains; by connecting the remaining low-admittance edges we form the boundaries between the subdomains. What remains is to find an expression for the admittance of a mesh edge.

First, examine the expression for charge, Q_b^i, transferred through an area equal to the cross section X_b of edge b_b at time t_i. It can be expressed as the sum of electron, hole, and displacement currents:

$$Q_b^i = \int_{t_i}^{t_{i+1}} \left[X_b \left(J_n|_b + J_p|_b \right) + \epsilon \frac{\partial E}{\partial t} \right] dt \tag{2}$$

Unfortunately, this expression is not appropriate to decompose the domain *a priori* since the currents, $J_n|_b$ and $J_p|_b$ depend on the operating point. Hence, a low-impedance edge could be overlooked as an area of localized charge transport if no current was flowing through it at time t_i.

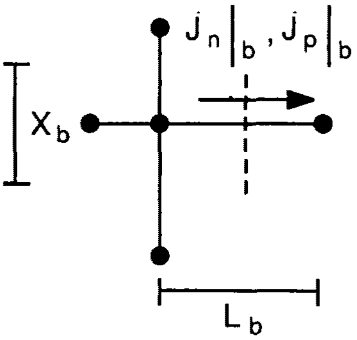

Figure 1: Terms defined on mesh.

Instead, we use the concept of charge transfer sensitivity. We can obtain from (2):

$$\frac{\partial Q_b^i}{\partial \psi_b} = \int_{t_i}^{t_{i+1}} \left[X_b \left(\frac{\partial J_n|_b}{\partial \psi_b} + \frac{\partial J_p|_b}{\partial \psi_b} \right) + \epsilon \frac{\partial}{\psi_b} \frac{\partial E}{\partial t} \right] dt \tag{3}$$

$$Y_b = \frac{\partial}{\partial V} \frac{\partial Q}{\partial t} = \frac{\partial}{\partial t} \frac{\partial Q_b}{\partial \psi_b} \tag{4}$$

where Y_b is the generalized mesh edge *admittance*,

In order to develop an expression for this impedance which can be easily used in an algorithm, we use the Forward Euler integration formula to integrate in time and we divide by h^i to represent time derivatives, where h^i is the time-step, $t_{i+1} - t_i$. We make a first order approximation to the spatial derivatives by dividing by L_b .

Under these assumptions, we obtain from (4):

$$Y_b = \frac{X_b}{L_b}(q\mu_n n_b^i + q\mu_p p_b^i + \frac{\epsilon}{h^i}) \tag{5}$$

Y_b represents a generalized mesh edge admittance where the first two terms are a conductance and the last one is a capacitance. Since in depletion regions the first two terms will be relatively small, their admittances will also be smaller in comparison to the channel and contact regions. Also, in oxide regions, the permittivity ϵ is lower than in silicon and a correspondingly lower admittance results. If an appropriate threshold value of the admittance, Y_{th} , is used to determine the subdomains, the likely places for boundaries to occur will be in the depletion regions or oxide regions. Note that this conclusion relates to perturbation analysis [11] where the slowly varying solution occurs away from regions of steep concentration gradient. These are regions where Y_b is low and where conduction current is not likely to flow.

Equipped with an expression for Y_{th} , the algorithm is fairly straightforward as shown in Figure 2. Any discritization which relates mesh nodes to mesh edges can be used. Mesh edges, b_b , with admittances greater than Y_{th} are grouped in nuclei by a crystalization-type process. Edges are sorted and selected in order of decreasing admittance and each is terminated by two nodes, N_1 and N_2 . When an edge is selected, if N_1 and N_2 are not in any nucleus, a new nucleus is formed and N_1 and N_2 are added to it. When a edge is selected and one of its nodes is already included in a nucleus, the other node is also added to this nucleus. Finally, if an edge is selected and its nodes belong to different nucleus, the nuclei and all nodes in them are merged. At the end of this process, we have clustered the regions of high-admittance into subdomain nuclei. We now form the boundaries between each subdomain.

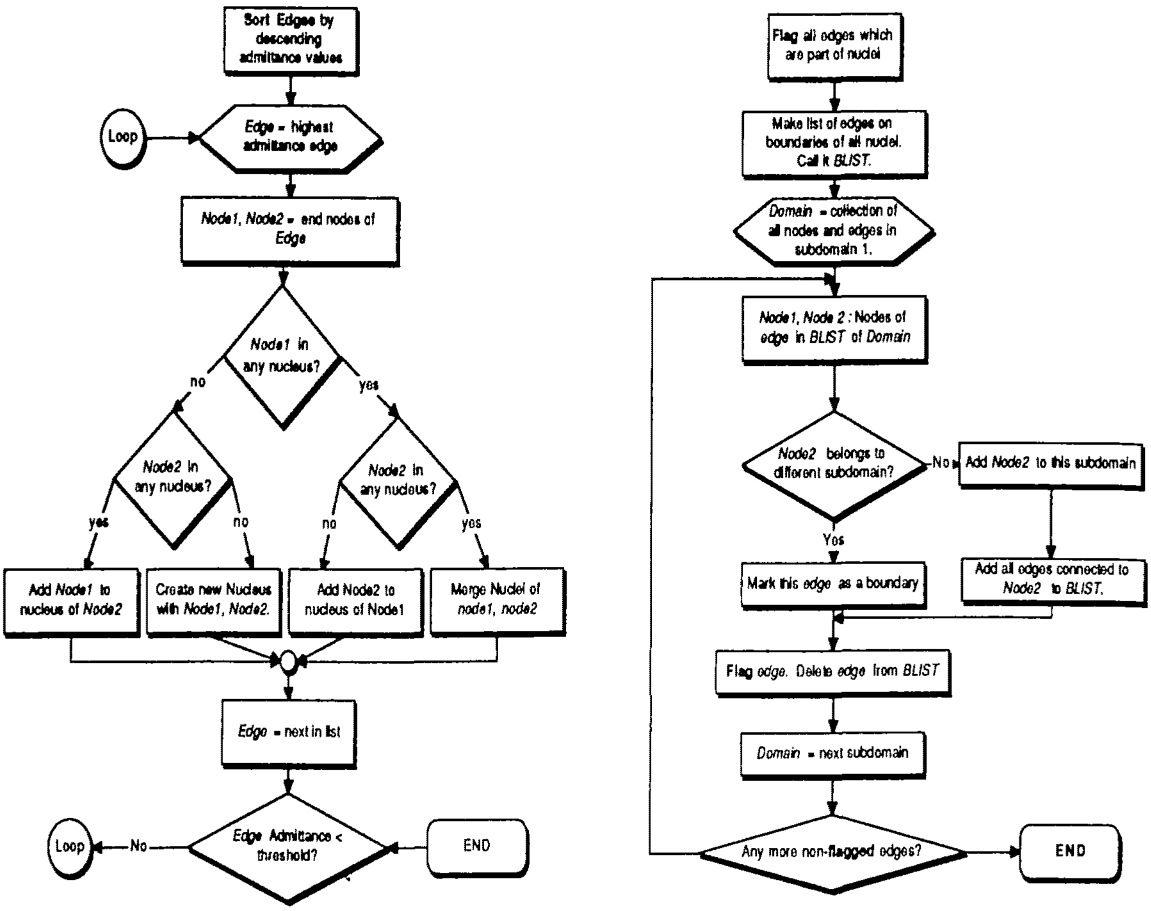
Sort Edges by descending admittance values
Loop
Edge = highest admittance edge
Node1, Node2 = end nodes of Edge
Node1 in any nucleus?
no
yes
Node2 in any nucleus?
yes
no
Node2 in any nucleus?
no
yes
Add Node1 to nucleus of Node2
Create new Nucleus with Node1, Node2.
Add Node2 to nucleus of Node1
Merge Nuclei of node1, node2
Edge = next in list
Loop
No
Edge Admittance < threshold?
END
Flag all edges which are part of nuclei
Make list of edges on boundaries of all nuclei. Call it BLIST.
Domain = collection of all nodes and edges in subdomain 1.
Node1, Node 2 : Nodes of edge in BLIST of Domain
Node2 belongs to different subdomain?
No
Add Node2 to this subdomain
Yes
Mark this edge as a boundary
Add all edges connected to Node2 to BLIST.
Flag edge. Delete edge from BLIST
Domain = next subdomain
Any more non-flagged edges?
END

Figure 2: Algorithms for Nuclei Formation and Border Formations

The edges on the outer boundary of each subdomain nucleus are first identified. An edge is selected from a subdomain boundary and all adjoining edges and nodes are then added to this subdomain. When an edge is selected where N_1 belongs to one subdomain and N_2 belongs to another, then a boundary is formed between N_1 and N_2 . The boundary line can be thought of as a perpendicular bisector to the mesh edge (as shown by the dotted line in Figure 1). The algorithm ends when all edges have been classified and all nodes belong to a subdomain.

The domain is now composed of subdomain nuclei, surrounded by low-admittance edges which form a border with other subdomains. This configuration isolates areas of localized charge transfer with equivalent nonlinear capacitive and conductive elements which control current flow between subdomain nuclei.

A critical point of the algorithm is the choice of Y_{th} . Whereas a low Y_{th} yields many subdomains (fine granularity) and a small sets of equations to solve for each, it introduces an unacceptably large border in the global block-bordered Jacobian matrix. Very coarse granularity provides little exploitation of the latency and limits flexible time-step control in the transient simulation. This tradeoff is explored in Section 5.

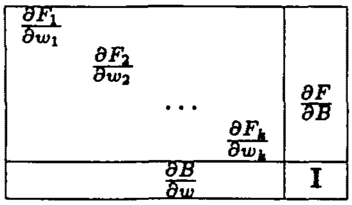
$\frac{\partial F_1}{\partial w_1}$
$\frac{\partial F_2}{\partial w_2}$
...
$\frac{\partial F_k}{\partial w_k}$
$\frac{\partial F}{\partial B}$
$\frac{\partial B}{\partial w}$
I

Figure 3: Bordered Block Diagonal Jacobian Matrix after Decomposition.

4 Solution Techniques - Matrix Formulation

The decomposed domain allows an efficient matrix formulation of the set of discretized nonlinear equations if we use the method of node tearing [12], which has been successful in attacking circuit problems of large dimension or with repeated circuit blocks [7], [8].

In addition to the normal device variables, we explicitly list the the electron and hole current flux and the electric field between subdomain boundaries (along subdomain boundary mesh edges) as variables. Naming these new variables, **B** , we have:

$$\mathbf{y} = (\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_n}, \mathbf{B})^T \tag{6}$$

$$\mathbf{w_i} = (\psi_1, n_1, p_i, \ldots, \psi m, n_m, p_m)^T \tag{7}$$

$$\mathbf{B} = \left(\hat{\mathbf{E}}_{1,2}, \hat{\mathbf{J}}_{n,1,2}, \hat{\mathbf{J}}_{p,1,2}, \ldots \hat{\mathbf{E}}_{j,k}, \hat{\mathbf{J}}_{n,j,k}, \hat{\mathbf{J}}_{p,j,k}\right)^T \tag{8}$$

where $\hat{\mathbf{E}}_{j,k}$, $\hat{\mathbf{J}}_{n,j,k}$, and $\hat{\mathbf{J}}_{p,j,k}$ represent the vectors of coupling variables between subdomain j and k. A block-bordered Jacobian matrix is thus formed as shown in Figure 3, where F_i represents the discretized semiconductor equations in subdomain i and where there are k subdomains. The diagonal submatrices represent the Jacobian for each subdomain while the border terms of the matrix incorporate all the coupling flux between subdomains. In addition to the ability to maintain better control of the convergence through such methods such as a Modified Two Level Newton Iteration [8], [7] [13], there is now a possibility in the transient analysis to re-use information of some diagonal blocks from previous time-steps.

If we use the one-step implicit time-integration algorithm based on exponential fitting: [14]:

$$\frac{\partial n}{\partial t} = f(\psi, n, p)$$

$$n_{i+1} = n_i + h[cf(\psi_{i+1}, n_{i+1}, p_{i+1}) + (1-c)f(\psi_i, n_i, p_i)] \tag{9}$$

we can vary the free parameter in the range $0.5 < c < 1.0$ for each domain, to minimize the truncation error while retaining the maximum time-step h.

The local truncation error of (9), can be expressed as:

$$E_i = h^2[\frac{1}{2!} - c]f'(x_i) + h^3[\frac{1}{3!} - \frac{c}{2!}]f''(x_i) + \cdots + h^n[\frac{1}{n!} - \frac{c}{(n-1)!}]f^{(n)}(x_i) \tag{10}$$

Based on these local truncation error estimates, a different h_k for each subdomain, SD_k, is found. Simulation *events* are scheduled at time t_i+h_k. When a simulation event occurs for SD_k, extrapolated estimates for the non-active subdomains can be used as the boundary conditions for SD_k which means using approximations for the Jacobian submatrix $\frac{\partial F_m}{\partial w_m}$ where($m \neq k$) instead of fully recomputing these submatrices.

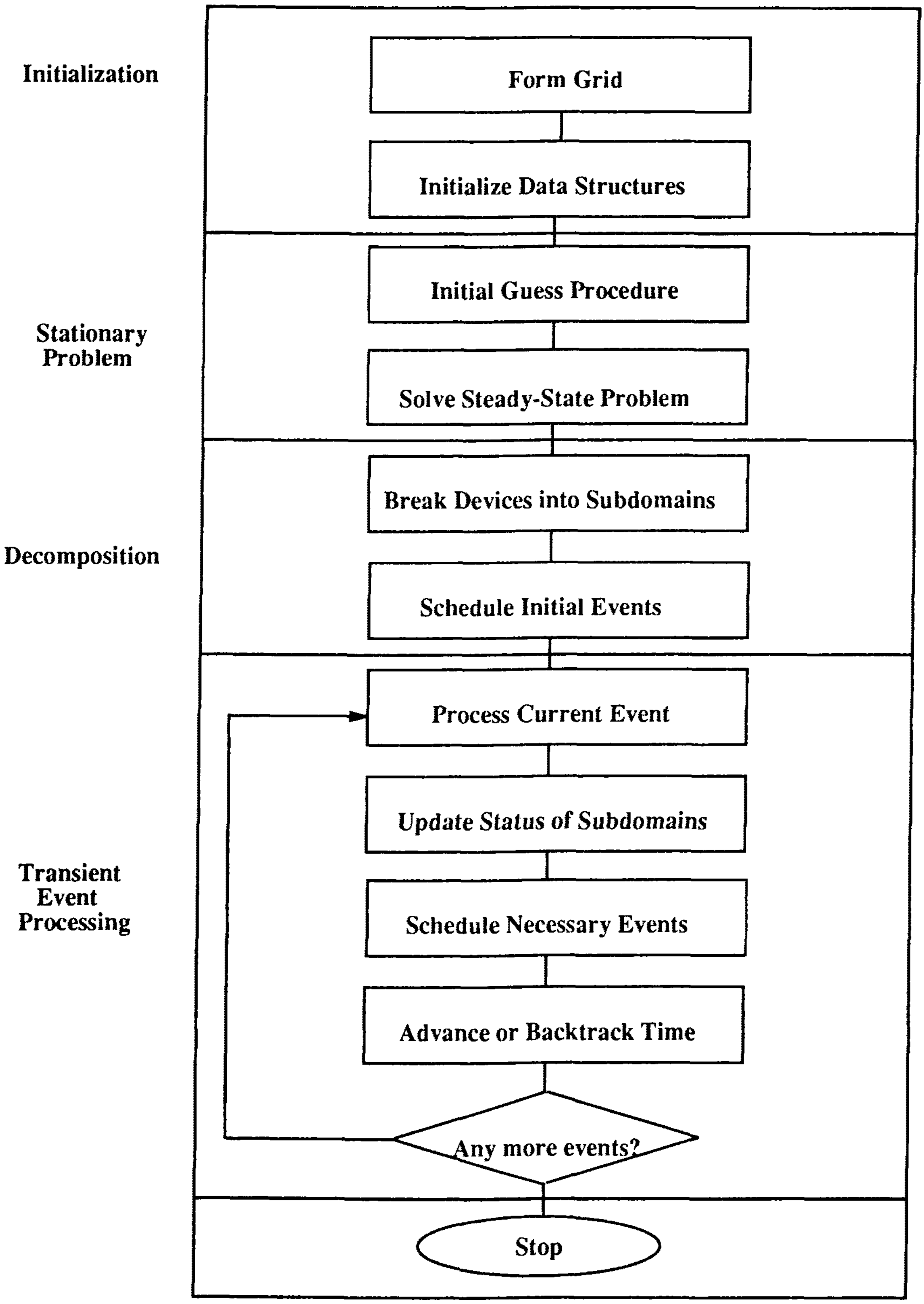
Initialization
Form Grid
Initialize Data Structures
Stationary Problem
Initial Guess Procedure
Solve Steady-State Problem
Decomposition
Break Devices into Subdomains
Schedule Initial Events
Transient Event Processing
Process Current Event
Update Status of Subdomains
Schedule Necessary Events
Advance or Backtrack Time
Any more events?
Stop

Figure 4: Flow of Transient Device Simulation.

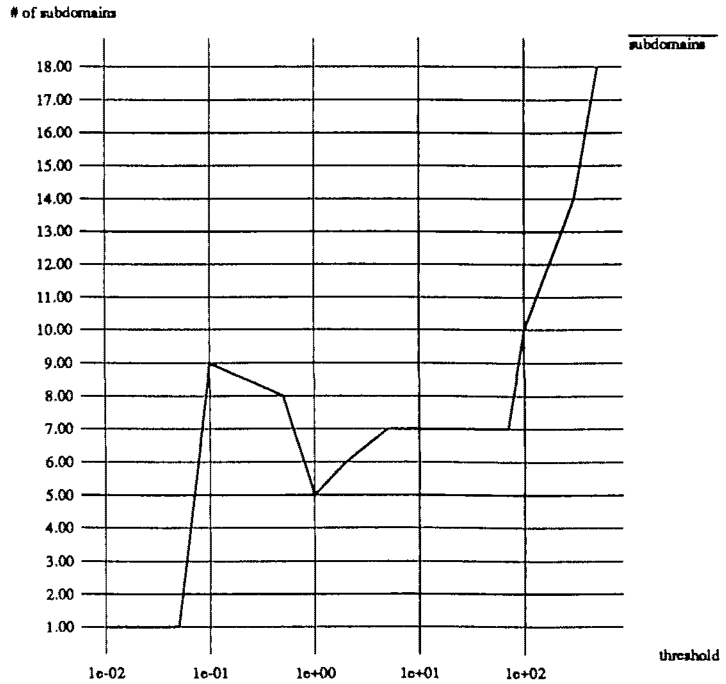

Figure 5: Graph showing dependence of domains on threshold value, Y_t.

This integration formula offers another opportunity to exploit the subdomain decomposition. If a set of subdomains are scheduled with events close together in time, these events can be merged and the free parameter c adjusted so that the estimated truncation error is minimized. This reduces some of the overhead of managing the events.

Because of the nonlinearity and time-dependence of Y_b , some subdomains may need to be merged together during a transient simulation, but this can be monitored by the charge transfer between subdomains which is explicitly represented in the solution vector.

A diagram of the flow of a transient simulation is shown in Figure 4.

5 Examples

The doping profiles and Poisson solution of a MOS transistor were obtained from MINIMOS [15] and were used to examine some characteristics of the decomposition algorithm. The mesh used for this purpose was generated by the simulator. Also note that a non-orthogonal mesh could have been used with similar results. In Figure 5 the best choice of Y_{th} occurs at the plateau in the graph. Selecting Y_{th} in this range yields a decomposition such that depletion regions and oxide regions become boundaries and we get a reasonable number of subdomains as shown in Figure 6. Note how the borders of the subdomains (indicated by a lack of mesh lines) occur in the depletion region indicating a charge transfer boundary as we would expect.

6 Conclusion

In this paper, we have shown an algorithm which can be used on general device geometries for decomposition into subdomains suitable for transient simulation utilizing latency. We have also provided the conceptual background for incorporating the decomposed regions into a transient device analysis utilizing event-driven simulation.

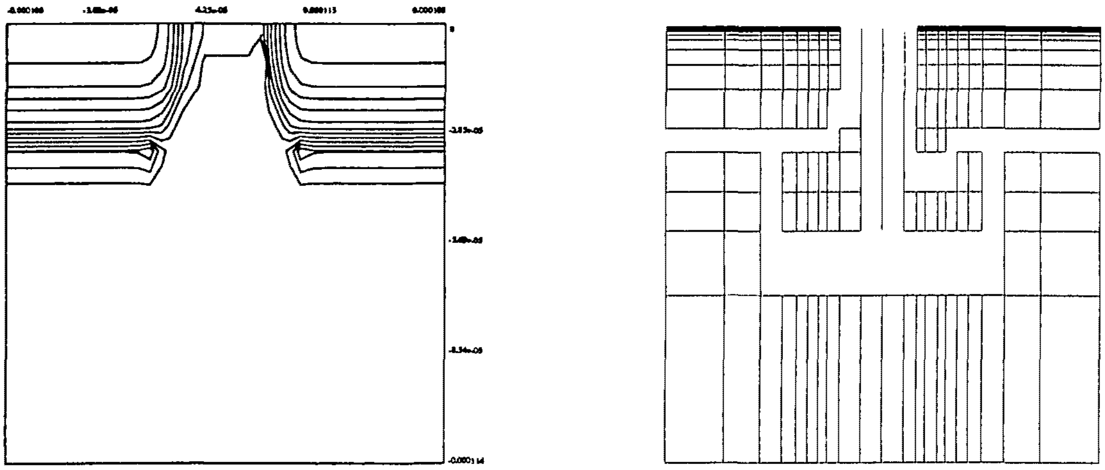

Figure 6: Doping profile and associated decomposition of a MOS transistor.

References

[1] R. E. Bank, J. William M. Coughran, W. Fichtner, E. H. Grosse, D. J. Rose, and R. K. Smith, "Transient simulation of silicon devices and circuits," *IEEE Transactions on Electron Devices*, vol. ED-32, pp. 1992–2006, October 1985.

[2] J.-H. Chern, J. T. Madea, J. Lawrence A. Arledge, and P. Yang, "Sierra: A 3-d device simulator for reliability modeling," *IEEE Transactions on CAD*, vol. 8, pp. 516–527, May 1989.

[3] J. S. Scroggs, *The Solution of a parabolic partial differential equation via domain decomposition: The Synthesis of Asymptotic and Numerical Analysis.* PhD thesis, University of Illinois at Urbana-Champagne, May 1988.

[4] M. J. Berger, "Adaptive mesh refinement for hyperbolic partial differential equations," *Journal of Computational Physics*, vol. 53, pp. 484–512, 1984.

[5] M. R. Dorr, "On the discretization of interdomain coupling in elliptic boundary value problems," in *Domain Decomposition Methods*, (Philadelphia, PA), Society for Industrial and Applied Mathematics, 1989, pp. 17 - 39.

[6] N. B. Rabbat and H. Y. Hsieh, "A latent macromodular approach to large-scale sparse networks," *IEEE Transactions on Circuits and Systems*, vol. CAS-23, pp. 745–752, December 1976.

[7] W. E. Engl, R. Laur, and H. K. Dirks, "Medusa - a simulator for modular circuits," *IEEE Transactions on Computer-Aided Design*, vol. CAD-1, pp. 85–93, April 1982.

[8] K. A. Sakallah, *Mixed Simulation of Electronic Integrated Circuits.* PhD thesis, Carnegie Mellon University, November 1981.

[9] T. Ohtsuki and K. Kani, "A unified modeling scheme for semiconductor devices with applications of state-variable analysis.," *IEEE Transactions on Circuit Theory*, vol. CT-17, pp. 26–32, February 1970.

[10] W. Gropp and D. E. Keyes, "A domain decomposition method with locally uniform mesh refinement," in *Domain Decomposition Methods*, (Philadelphia, PA), Society for Industrial and Applied Mathematics, 1989, pp. 115–129.

[11] S. Selberherr, *Anaylsis and Simulation of Semiconductor Devices.* Wien: Springer-Verlag, 1984.

[12] G. Kron, *Tensor Analysis of Networks.* Wiley, 1939.

[13] K. Mayaram, *CODECS: A Mixed Level Circuit And Device Simulator.* PhD thesis, University of California, Berkeley, November 1988.

[14] J. D. Lambert, *Computational Methods in Ordinary Differential Equations.* London, New York, Sydney, Toronto: John Wiley & Sons, 1973.

[15] S. Selberherr et. al., *MINIMOS 5.0.* Institute of Microelectronics, Techical Univeristy of Vienna, 1990.