Multiple Zone Inverse Diffusion Solver for Silicon Processing

Y. S. Wang and C. H. Mastrangelo
Center for Integrated Sensors and Circuits
Department of Electrical Engineering and Computer Science, University of Michigan
Rm. 2405, EECS Bldg., Ann Arbor, MI 48109-2122, USA
(313)-763-7162, *FAX*:(313)-763-9324

***Abstract*— In this paper we present the implementation and testing of an inverse diffusion solver that determines implant doses and drive-in times for a set of coupled one-dimensional diffusion profiles spreading over several regions of a device. The solver starts with a description of each distinct device diffusion region in terms of a set of junction depth x_j and peak concentration c_s specifications, and a simplified description of the process flow. From this information, the solver setups a storage data structure containing the full history of each profile within each region at each process step. Starting from a guess for the dose and drive-in times, the software first fills the storage structure with approximate profiles and then iteratively solves the problem backwards starting from the last diffusion to the first followed by a forward guess correction loop. The backward-forward recursion loop is repeated until errors in x_j and c_s are negligible. The solution method was successfully tested with several conventional device structures.**

I. Introduction

The calculation of diffusion profiles in semiconductor devices using software simulation plays an important role in silicon processing. Most diffusion solvers require a process with fixed doses, temperatures, and drive-in times as input yielding corresponding dopant profiles as outputs. This calculation procedure is a consequence of the numerical methods used for the solution of the partial differential equations dictating the dopant motion. However in device process development, designers are often faced with the inverse problem of calculating these diffusion parameters necessary to achieve specific profile requirements. This type of inverse problem is often solved in an iterative fashion or through the use of numerical optimization procedures that "zero-in" the correct diffusion parameters [1]. The problem becomes more difficult if there is more than one profile involved. In complex device structures such as BiCMOS, there is coupling between seven or eight diffusions; hence their parameters must be solved simultaneously. Furthermore, the coupling is intrinsically connected with the process flow which determines at which time dopants are introduced in distinct regions of the device.

In this paper we present a method of solution for this type of coupled inverse problems. In this method, the device is first divided onto a set of distinct diffused regions or "zones" as shown in Fig. 1. Each of these zones is considered as a one-dimensional problem. The coupling of one zone to another takes place through causality in the process steps or physical overlap of individual diffusions. In this solver, all diffused profiles are accepted in terms of a final junction depth x_{jf}, peak concentration c_{sf}, and corresponding zone. The solver uses a description of the process flow as input and calculates the implant dose Q and drive-in time t for all diffusion steps to achieve desired profile specifications at the end of the process. This

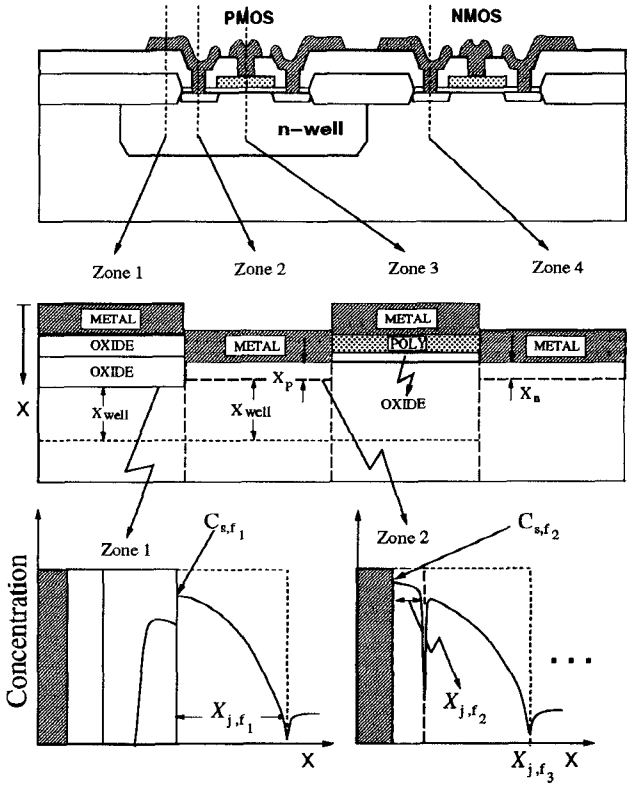

Fig. 1. the implant dose Q and drive-in time t can be solved by giving the value of junction depth $x_{j,f}$ and peak concentration $c_{s,f}$ of each dopant profile in each zone.

multizone inverse diffusion solver is one of the submodules used in the process compiler MISTIC [2] which accepts a device cross section as input and generates process flows as outputs.

II. Problem Definition and Domain Partitioning

In the multizone approach, the concentration of dopant species C_i in each of the zones is regulated by the well

known nonlinear PDE [3], [1]

$$\frac{\partial C_i}{\partial t} = \nabla \cdot D_i(\nabla C_i + Z_i f_i \nabla C_{net}) \quad (1)$$

where f_i is the electric field enhancement factor

$$f_i = \frac{0.5N_i}{[(0.5C_{net})^2 + n_i^2]^{0.5}} \quad (2)$$

and

$$C_{net} = -\sum_{i=1}^{n} Z_i N_i \quad (3)$$

where N_i is the electrically active concentration of the i^{th} impurity. In order to find doping profiles, Eq. (1) must be solved subject to a series of initial conditions and thermal cycles. The initial conditions are determined by a set of ion implants that introduce the dopant into the solid and the profile history at different steps in the process. Hence, the implant doses and drive-in cycles (Q, t) in a region must be known to solve for C_i which ultimately determine x_j and c_s. The inverse problem therefore requires the use of an iterative procedure.

Starting with a guess (Q_n, t_n) for the dose and time, Eq. (1) is solved yielding an approximate $(x_{j,n}, c_{s,n})$ and error $\vec{\epsilon}_n = (x_{j,n} - x_{j,f}, c_{s,n} - c_{s,f})$. This error can then be used to correct the initial guess yielding (Q_{n+1}, t_{n+1}) and so on. The calculation of C_i is computationally expensive; therefore it is important to select an iterative scheme that requires a minimum number of profile calculations.

The coupling of profiles within different zones takes place temporally in the sense that all profiles present in the device at a given time are affected by subsequent drive-in cycles. Furthermore, it is possible to overlap diffusions over several zones. In this case, several zones are affected by the same implant, and the specifications on this type of diffusion can only be met in a single zone. The solution scheme described here accounts for these interactions.

A. Internal Data Storage

In order to keep track of multiple profiles in several device zones at different times in the process, the solver setups the six level profile storage data structure shown in in Fig. 2. At each step in the process, the device contains several diffused zones. Each zone contains several profiles, each profile corresponding to a different impurity. Each of these profiles may be distributed among several layers, and each layer contains a one dimensional mesh consisting of a linked list of a variable number of elements. Each of these elements contains individual nodes where individual concentrations are stored. The data structure is setup to store and launch numerical finite element solutions of Eq. (3) at any step during the process.

The highest level step structure stores step type and parameter information including type of impurity, current dose and drive-in cycle time, temperature, current and final junction depth and peak concentration. Some of this information is copied to the profile structures below.

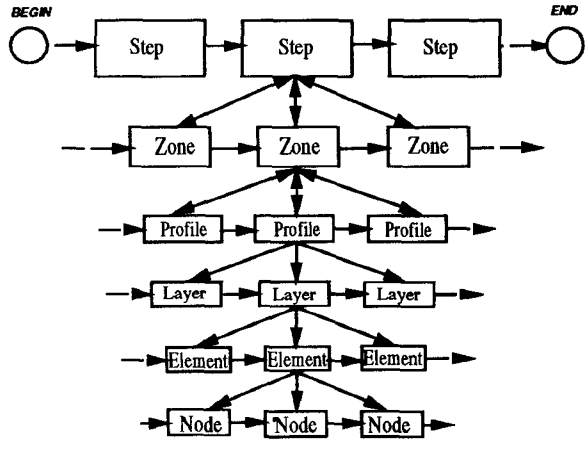

Fig. 2. Internal data structure tree of multizone inverse diffusion solver

B. Finite Element Profile Solution

Eq. (3) is solved using the method of finite elements. We have adopted this scheme because it is easily expanded to higher dimensions. Alternatively, Eq. (3) could be solved with a conventional diffusion solver such as SUPREM [1]. Equation (3) is first discretized over each element and translated into the nonlinear matrix system

$$A(\alpha)\alpha + B\dot{\alpha} = P, \quad (4)$$

where $A(\alpha)$ is the global stiffness matrix, and B is the global capacity matrix, and α is the solution vector. P is a vector describing the flux of impurities leaving or entering the device domain. The calculation of these matrices is an standard procedure described in many textbooks [4].

This first order nonlinear time dependent system of differential equations is known to be stiff and can be solved numerically in several ways. One of the most common schemes solves for α at discrete times using the modified backward difference method (MBD) [5]

$$\hat{A}_{n+1}\alpha_{n+1} = B\alpha_n, \quad (5)$$

where $\hat{A}_{n+1} = A(\alpha_n) \triangle t + B$ and $\hat{B} = B$ which can be solved iteratively to update $\hat{A}_{n+1}$ with the new α_{n+1}. Culham [5] showed that this scheme is convergent in just a few iterations.

A drawback of the MBD method is that the error is linearly dependent on the time step giving $\epsilon \propto \triangle t$. A more precise numerical method of solution for Eq. (4) employs the predictor corrector method [6]. The predictor corrector (PC) method uses two steps to get solutions with error $\epsilon \propto \triangle t^2$. In this scheme, Eq. (4) is solved using an intermediate solution vector α^*_{n+1}

$$\begin{aligned}(A(\alpha_n) \triangle t + B)\alpha^*_{n+1} &= B\alpha_n \\ (A(\alpha^*_{n+1}) \triangle t + B)\alpha_{n+1} &= B\alpha_n. \end{aligned} \quad (6)$$

Our numerical experiments predicted a more accurate result from the PC method, but it is more computationally expensive than the MBD method since it requires two steps per iteration. The FEM solver is implemented using a fully-adaptive mesh that adds or removes nodes from each node as the shape of each profile changes.

III. Inverse Solution Method

In the most general sense, this inverse problem could be considered as a $2N$ multivariable optimization problem since for each diffusion in the device, two parameters (Q,t) must be determined subject to two specifications in $(x_{j,f}, c_{s,f})$. This multidimensional view however suffers from many drawbacks. First, the problem can have many local minima, and its picture is obscured by the high dimensionality. Furthermore, each function evaluation is very computationally expensive because each point requires many FEM solutions.

A great deal of computational savings can be achieved from the following observation. Since the dopants are introduced into the device in a known order, it is possible to minimize the amount of computation by solving for the parameters for the last diffusion first. This is justified because the motion of dopants in the last diffusion is only influenced by the last drive-in cycle and the existent profiles. Since the existent profiles were subject to many drive-in cycles, they are in general well developed at this stage; hence in general they are minimally affected by the last drive-in cycle. Therefore the parameters for the last diffusion are largely determined by an approximately stationary background and the drive in time for the last thermal cycle followed by the diffusion prior to the last and so on. The main virtue of this scheme is that at any given time, parameters for a single diffusion are determined.

Essential to this scheme is the availability of a good initial guess, and an iterative scheme to correct it. In our scheme we first determine initial guesses for (Q,t) assuming that the profiles are approximately gaussians, and implants are considered to be shallow. Junction depths are computed from intersections between two gaussians or a gaussian and a constant background. The junction depth constraints form a set of equations yielding the final straggle, $(\beta_T)_i$, for each profile which is the cumulative Dt product of all successive thermal steps. A simultaneous solution results in the matrix equation $\tilde{\alpha}\,[\widehat{T}]\,\widehat{\beta}_p + \widehat{C} = \widehat{\beta}_T$ where $(\beta_p)_i = D_i(T_i)t_i$ are partial contributions due to each variable drive-in, and $\hat{C}$ accounts for all fixed thermal cycles. Elements in matrix $\tilde{\alpha}$ are diffusivity ratios $\alpha_{ij} = D_i(T_j)/D_j(T_j)$.

The matrix equation is initially solved with a fixed temperature. Our implementation uses an initial choice of 1000°C for all steps. Since the i^{th} diffusion is only affected by subsequent steps $j \geq i$, then matrix $\tilde{\alpha}$ is upper triangular and solved easily by back substitution. If the drive-in times are too long or too short, T_i is increased or decreased correspondingly to fit an acceptance time window. If all $\beta_p \geq 0$, corresponding drive-in times and implant doses are calculated from

$$\begin{aligned} t_i &= \frac{(\beta_p)_i}{D_i(T_i)} \\ Q_i &= \sqrt{\pi(\beta_p)_i}\, c_{s,i} \end{aligned} \tag{7}$$

The gaussian derived estimates for (Q,t) give us a rough idea of the required doses and drive-in times, but in general they will not differ from the actual values by more than an order of magnitude. The diffusion solver thus starts with a Gaussian based guess for the profiles and iteratively refines this guess until sufficient convergence is achieved unique solution.

In this scheme, Q and t are solved using a recursive backward loop with non-causal corrections in a forward loop as shown in Fig. 3. The dose and diffusion time

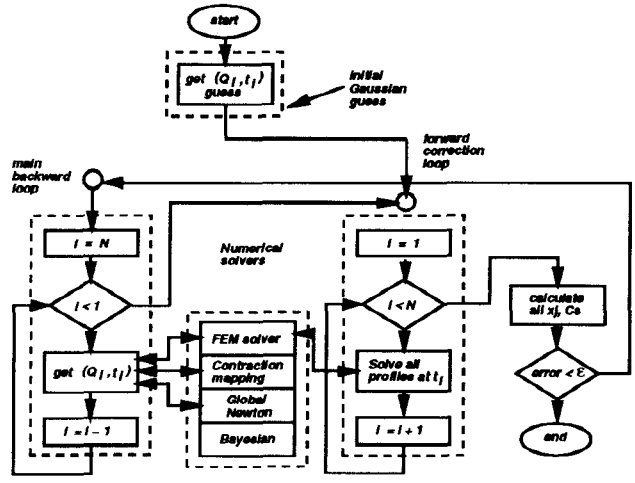

Fig. 3. **Structure of inverse diffusion solver**

(Q_{k-1}, t_{k-1}) for the last k^{th} diffusion are solved numerically to conform to $x_{j,k}$ and $c_{s,k}$ specifications. Next, the $k-1$ diffusion parameters (Q_{k-2}, t_{k-2}) are solved with the updated (t_{k-1}, Q_{k-1}) and corresponding numerical profile. The backward recursion is continued until (Q_o, t_o) is found. In the forward loop, new numerical guesses are calculated using the updated (Q_j, t_j) to reduce the error in the next backward loop. These new guesses are used again in the backward loop to obtain second order corrections. The backward-forward recursion loop is repeated until errors in x_j and c_s in the forward loop are negligible.

Numerical solutions of the diffusion PDE provide (x_j, c_s) in terms of (Q_j, t_j), but not vice versa. Therefore the correct (Q_j, t_j) are arrived through iterative error minimization in $\epsilon_{x_j} = (x_j - x_{j,f})$, and $\epsilon_{c_{s,j}} = \log(c_{s,j}/c_{s,f})$, where $(c_{s,f}, x_{j,f})$ are its desired values. Several iteration schemes for (Q_j, t_j) have been implemented. These include a contraction mapping, a globally convergent Newton method, and a Bayesian global optimization [7] in that order. In general, approximately 5-10 iterations are required for convergence in each diffusion. Typically 5-25 backward-forward loops are necessary for full convergence within 3 % of specifications. Since the existence of solutions is not warranted, if the first two algorithms fail, the Bayesian global optimizer finds the best possible fit.

The most efficient method of iteration is the contraction mapping scheme. In this scheme the corrections in (Q,t) are directly related to the error in (x_j, c_s) by the following relation

$$t_{n+1} = t_n \left(\frac{x_{j,f}}{x_j}\right)^2 \quad , \quad Q_{n+1} = Q_n \left(\frac{c_{s,f}\, x_{j,f}}{c_{s,j}\, x_j}\right) \tag{8}$$

Eq. (8) is in general a contraction which converges very quickly. The main difficulty with the method is that it is not always convergent, specially for diffusions that have very high concentrations. Currently, a number of other more promising schemes are being attempted. These methods are based on finding a good "fitting" function for the global error that can be evaluated quickly [8].

IV. Testing

The solution method was tested with several structures. Figure 4 shows a table comparing the performance of the MBD and PC schemes when tested with a Sun UltraSPARC station. Both methods require roughly the same number of matrix iterations.

	2 zone 2 diffusions		3 zones 3 diffusions		4 zones 4 diffusions	
	iteration	time	iteration	time	iteration	time
MBD	162	0.38 min.	276	0.65 min.	384	0.83 min.
PC	164	0.65 min.	246	1.05 min.	354	1.42 min.

Fig. 4. **comparison of the performance of the modified backward difference and predictor and corrector scheme used in inverse diffusion solver**

The diffusion solver was then tested with a CMOS and a BiCMOS structure. Fig. 5 shows comparison of doping profiles in the CMOS devices at the source region with those obtained from SUPREM III using the parameters determined by the solver. Figure 6 shows the same comparison with an 8-simultaneous diffusion, 4-zone, twin-well BiCMOS structure. The diffusion solver was able to find the implant doses, drive-in temperatures and times to meet the specified junctions and peak concentrations within 2 % of specifications. The small difference in the

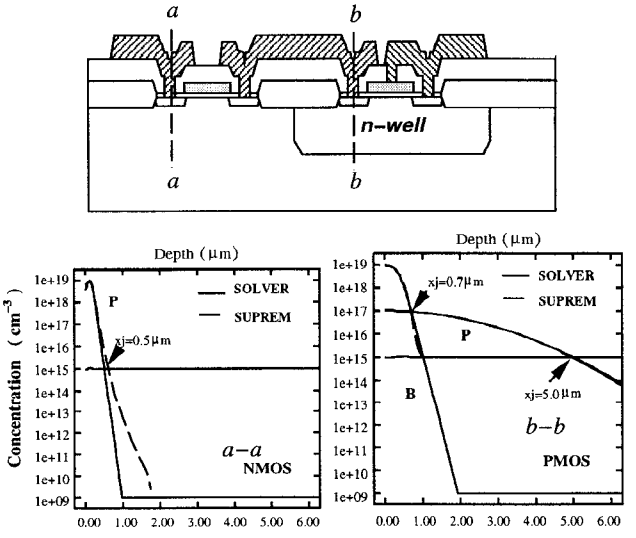

Fig. 5. **CMOS S/D profiles**

diffusion profiles is a consequence of slightly different diffusion coefficient models. The BiCMOS problem required roughly 1000 iterations.

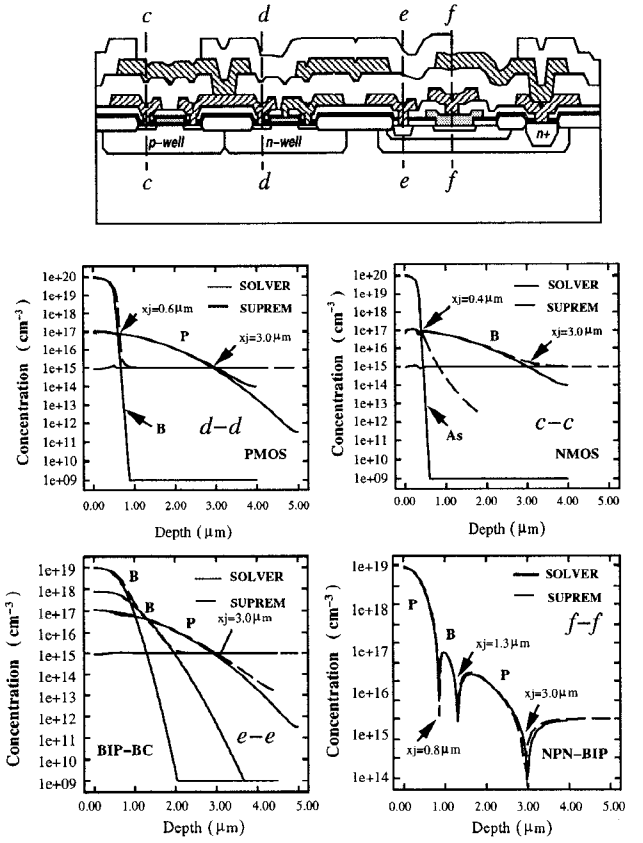

Fig. 6. **BiCMOS S/D and BIP profiles**

V. Summary

The testing results show that the inverse diffusion solver can estimate accurately the dose Q and drive-in times t for multiple profile requirements in the same device. Using this scheme, a process designer can determine diffusion parameters invoking the solver only once. This solver is included as a submodule of the process compiler MISTIC.

References

[1] C. Ho, J. Plummer, S. Hansen, and R. Dutton, "VLSI process modeling - SUPREM III," *IEEE Trans. Electron Devices*, vol. ED-30, pp. 1438–1453, 1983.

[2] M. Hasanuzzaman and C. H. Mastrangelo, "Process compilation of thin film microdevices," *IEEE Transactions on Computer-Aided Design of Integrated Circ uits and Systems*, vol. 15, no. 7, July 1996.

[3] B. Penumalli, "A comprehensive two dimensional vlsi process simulation program, BICEPS," *IEEE Trans. Electron Devices*, vol. ED-30, pp. 986–992, 1983.

[4] J. Reddy, *An Introduction to the Finite Element Method.* New York: McGraw-Hill, 1985.

[5] W. Culham and R. Varga, "Numerical methods for time-dependent, nonlinear boundary value problems," *Journal of Society of Petroleum Engineers*, 1970.

[6] W. Ames, *Numerical Methods for Partial Differential Equations.* Boston: Academic Press, 1992.

[7] J. Mockus, *Bayesian Approach to Global Optimization : Theory and Applications.* Boston: Kluwer, 1989.

[8] J. W. Bandler, R. W. Biernacki, S. H. Chen, and R. H. Hemmers, "Electromagnetic optimization exploiting aggressive space mapping," *IEEE Trans. Microwave Theory Tech.*, vol. 43, pp. 2876–2882, 1995.