VISTA Status Report
June 2001

T. Binder, J. Cervenka, A. Gehring, C. Harlander,
C. Heitzinger, S. Selberherr

Institute for Microelectronics
Technische Universität Wien
Gusshausstrasse 27-29
A-1040 Vienna, Austria

Contents

1 A Comparative Study of Two Numerical Techniques for Inductance Calculation in Interconnect Structures

1.1 Introduction

One consequence of technology scaling by shrinking feature sizes and increasing clock frequencies is the growing importance of interconnect lines. The performance of interconnects is limited by various parasitic effects (eg. signal delay, capacitive and inductive crosstalk, attenuation). The utilization of new materials (Copper and low-k dielectrics) reduces the RC time constant. Thereby decreased resistance and capacitance bring out inductive effects more intensively, requiring consideration in circuit simulation. Thus, inductance extraction becomes necessary for critical nets.

1.2 Physical approach

We compare two stationary inductance calculation methods both based on a numerical solution of Neumann's formula [1] for a precalculated current density distribution:

$$L_{ik} = \frac{\mu}{4\pi}\frac{1}{I_i I_k}\int_{V_i}\int_{V_k'}\frac{\vec{J}_i(\vec{r}\,)\cdot\vec{J}_k(\vec{r}\,')}{|\vec{r}-\vec{r}\,'|}\,\mathrm{d}V\,\mathrm{d}V'\ . \tag{1}$$

The integration is carried out numerically, where special attention has to be paid on the singularities of the integrand, or with the Monte Carlo method. For both methods the stationary current density is calculated with the finite element method. The first method [2] employs a summation of the contributions of all pairs of finite elements to solve the integral (Eq. 1), where different kinds of approximation are used, depending on the term $|\vec{r}-\vec{r}\,'|$. For a large distance (compared to the tetrahedron diameter) simple integration formulae are sufficient. The evaluation for small distances demand special formulae with certain integration points, published by Stroud [3] who presented various integration formulae which are applicable for various n-simplexes (e.g. the unit triangle, the unit tetrahedron) as integration region. If $\vec{r}$ and $\vec{r}\,'$ are in the same tetrahedron, a partially analytic integration scheme is used to increase the accuracy of the integration.

1.3 The Program Package

The SMART ANALYSIS PROGRAMS [4] uses the finite element method, because of it's advantages, as numerical robustness, the ability to solve nonlinear systems, high obtained accuracy, and general applicability.

The geometry can be defined either directly from the layout by specifying layer thicknesses, or by a rigorous topography simulation [5, 6]. The layout of the interconnect structure can also be imported from CIF or GDSII files, or created interactively with a graphical layout editor [7]. Furthermore, the program package includes three preprocessors, one for two-dimensional applications (CUTGRID) the other for three-dimensional applications. The preprocessor LAYGRID allows a layer-based input of the simulation geometry and the specification of the boundary conditions on the borders of each subdomain. The fully unstructured three-dimensional Delaunay grid generator DELINK [8] utilize an advanced-front algorithm, whereby the mesh generation starts from the initial front to fill up the solids with tetrahedrons.

A preconditioned conjugate gradient solver (ICCG), which has been optimized specifically for the discretized Laplace operator, is used to solve the linear systems for domains of conducting materials [9]. By applying Ohm's law to the derivative of the electrostatic potential the distribution of the electric current density is obtained. The simulation is performed with the module STAP (Smart Thermal Analysis Program), where both inductance extraction methods have been implemented.

Two postprocessors complete the program package, whereby the visualization tool SV is based on VTK [10], a flexible and powerful visualization library. Both postprocessors can be used to verify the grid quality, and for the visualization of several distributions (e.g. electric potential, temperature, current density), whereby SV provides numerous features, as eg. cutting plains, volume rendering and contour faces representation of distributions. Fig. 1 gives an overview about the SMART ANALYSIS PROGRAMS.

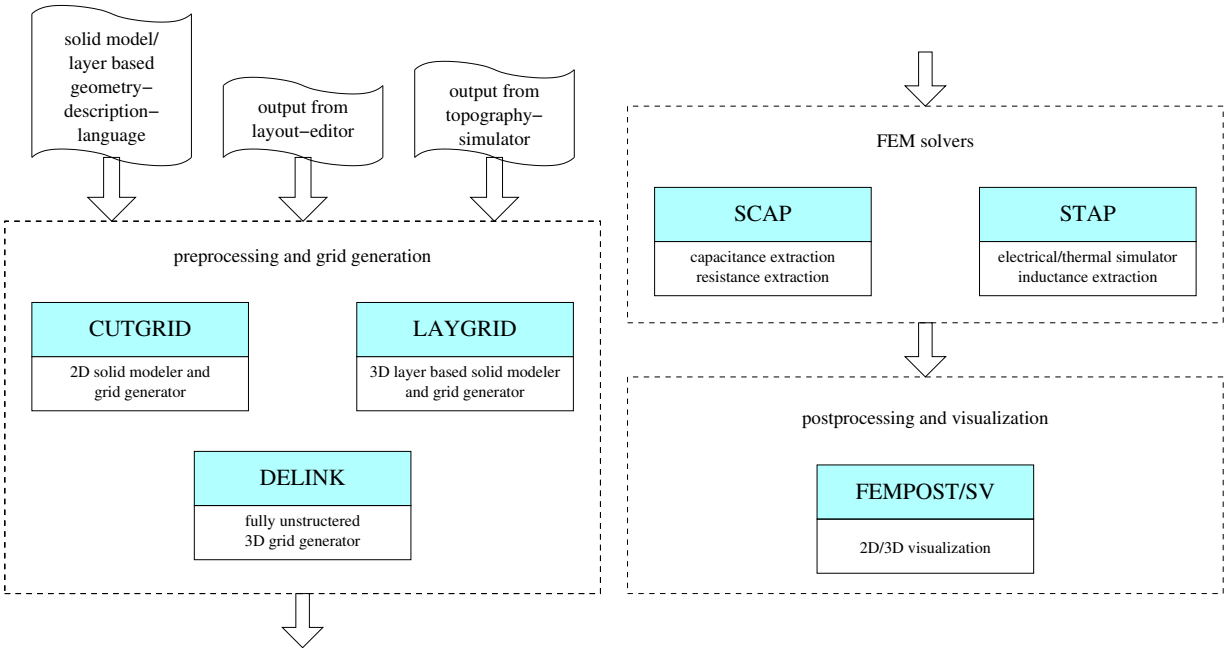

Figure 1: The SMART ANALYSIS PROGRAMS: tools and dataflow

1.4 The Monte Carlo Implementation

A well-known choice for the evaluation of multiple integrals is to apply the Monte Carlo method. Associated with this method, where by random the point coordinates are chosen, is a fairly high effort on CPU-time, because of the time consuming search for the associated element of the random point coordinates. To reduce the error a high number of function evaluations has to be carried out, whereby for each evaluation the aligned element with the precalculated current density must be found. To improve the convergence during the Monte Carlo sampling several variance reduction schemes (e.g. importance sampling, control variates) are known to accelerate the computation procedure [11].

One big advantage of our implementation is to bypass the high computational effort for the element location. We first determine the associated element to the evaluated probability function, and then locate the point inside the tetrahedron. For this purpose we take two arrays for every conductive segment. In the first one is the volume of each element, whereby the sum of all entries is scaled to one. In the second one is the probability function already evaluated for each conductor element by adding up all entries from the beginning to the current index of the first array. Then the random generator chooses a number between zero and one. The associated element complying to the probability function is found by a binary search.

To ensure a uniform probability the local coordinates of the integration points are found by shooting into the unit cube. The first point inside the registered unit tetrahedron is taken. For the interpolation of the current density inside each element quadratic shape functions are used.

1.5 Application Example

Fig. 2 and Fig. 3 show the current density of two planar transformers. These transformers are build of two interwound spirals each of 3, respectively, 5-turns metal with $5\ \mu m$ width, a spacing of $15\ \mu m$, and an inner length of $54\ \mu m$.

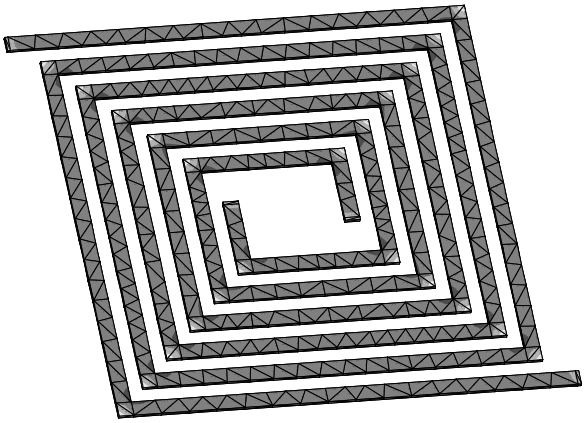

Figure 2: Current density distribution of the planar transformer with 3-turns metal

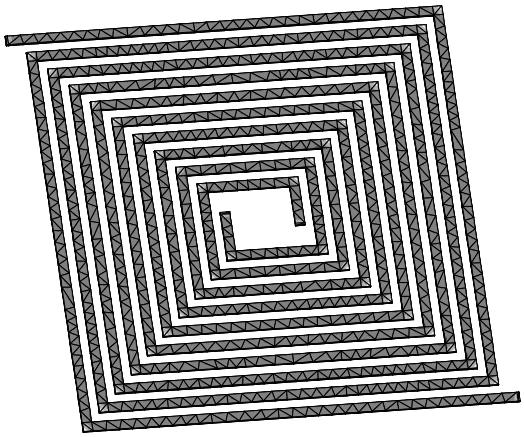

Figure 3: Current density distribution of the planar transformer with 5-turns metal

By utilizing the preprocessor LAYGRID three different grids were made. In Table 1 the simulation times for the current density and the Monte Carlo method, respectively, the first numeric method as accomplished above, and the calculated inductances are listed. The simulations were performed on a Digital Alpha workstation (DEC600/333 MHz). The number of samples for the Monte Carlo method was 1 million. The first column of Table 1 implies all elements of the conductive segments, whereby tetrahedral grid elements with quadratic shape functions were used. The analysis time for the Monte Carlo method is not so strongly influenced by the number of elements (n), because the computational effort for the binary search grows with ln(n). The simple integration formulae for the mutual inductances demand with increasing n almost the same time. Table 1 emphasize the advantages of the Monte Carlo method explicitly.

	Number of Elements	Time [s]				Results [nH]			
		MC		Method [2]		MC		Method [2]	
		M	L	M	L	M	L	M	L
3-turns	1800	17	33	1	327	0.67	1.04	0.67	1.06
	1968	17	34	1	627	0.67	1.05	0.67	1.06
	2648	18	34	3	1764	0.67	1.06	0.67	1.06
5-turns	4383	18	35	7	1945	2.71	3.58	2.70	3.62
	4653	19	35	8	2088	2.71	3.60	2.70	3.62
	5697	19	36	15	7885	2.70	3.60	2.69	3.63

Table 1: Analysis time and results of the planar transformers

1.6 Conclusion

We have presented a comparative study of two numerical techniques for inductance calculation in interconnect structures. Both methods are implemented into the package SMART ANALYSIS PROGRAMS, which allows simultaneous extraction of three-dimensional effective parameters of VLSI circuits.

2 Optimization of Industrial High Voltage Structures by Three-Dimensional Diffusion Simulation

2.1 Introduction

The requirement for a low switch on resistance (R_{on}) is to design a single device as small as possible. The reduction of space charge regions is limited by the dopand surface concentration of the wells which may result in impact ionization effects in case of too high doping concentrations. On the other hand, lowering the doping concentrations is limited by the required punch-through voltage. To fulfill these conflicting criterions the doping concentrations must be optimized.

2.2 Simulated structure

The investigated device is the tip of a drain finger of a high voltage PMOS transistor. To optimize the two-dimensional PMOS transistor it is necessary to ensure that no three-dimensional effects dominate the device behavior. The drain finger is implanted using a PTUB mask. To obtain proper electrical isolation from the wafer substrate, the PTUB is located in a shallow NWell (SNTUB) deep NWell (DNTUB) combination (SDNTUB). The PTUB, SDNTUB and the substrate form a pnp structure and under normal operation the PTUB/SDNTUB junction is biased in reverse direction. The optimal drain finger layout ensures that when applying maximal V_{dd} no punch-through between PTUB and substrate happens and no avalanche breakdown occurs at the surface of the wells. The complete device is embedded in the SNTUB so that there is no direct connection between PTUB and substrate. Only in the area of the PTUB, the DNTUB determines the distance between the pn- and the np-junctions. The PTUB/DNTUB mask layout is given in Fig. 4, which shows that the DNTUB mask is enclosed by the PTUB mask. To enlarge the distance between the two junctions it is necessary to use a long DNTUB diffusion time so that at the tip of the drain finger the DNTUB dopands nearly diffuse spherically. This long DNTUB diffusion finally leads to a DNTUB formation which starts outside of the PTUB mask. The three-dimensional consideration is necessary because the spherical diffusion of the DNTUB dilutes the DNTUB concentration in the area of the finger's tip and thus reduces the punch-through voltage of the PMOS device.

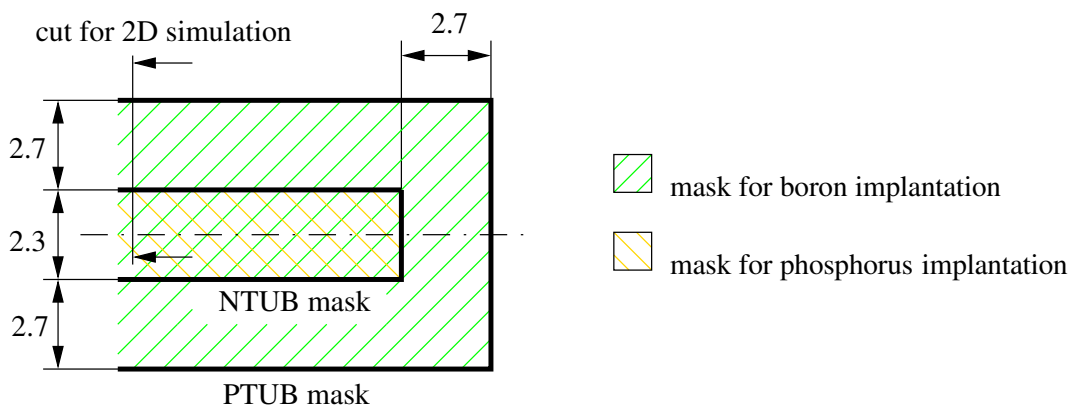

Figure 4: Well mask layout, units in μm

2.3 Comparison of simulating approaches

The conventional procedure is to simulate the whole ion implantation process first [12] and then the three-dimensional transient diffusion [13]. Thereby both steps require a particularly refined grid to achieve appropriate accuracy [14] and, therefore, the vast amount of memory and huge calculation times constitute prohibitive demands in practice.

Alternatively we have chosen a method, specially adapted to this problem. Because of the long diffusion ranges, the exact simulation of the ion implantation process can be neglected and the implanted ions were assumed as Dirac impulses only located at the wafer top. With this simplification the final diffusion profile inside the wafer can be calculated as the sum of some partial diffusion processes, represented by the Green's function of the diffusion equation [15].

2.4 Calibration and evaluation

Attention must be put to preserve the dose of the implanted ions and therefore a dose integration after implantation and after simulation must be carried out. In addition these models have to be calibrated by the two-dimensional simulation results which are available far away from the tip of the finger. For the full three-dimensional simulation a sufficiently fine grid in the areas of high diffusion gradients must be granted and therefore the simulation time was enormous, whereas for the simplified algorithm a grid is only necessary at the surface of the wafer and the resulting doping distribution can be calculated at any point of interest. It is to note that a limitation of this method is obviously given, if the ranges of the implantation depth and diffusion width get in the same size.

The assessment criterion of the new layout parameters is the fact that the dopant concentration of the PTUB/SDNTUB junction at the surface of the wells is the same for the two-dimensional case and the three-dimensional finger case. This ensures that the breakdown at the surface in the three-dimensional structure takes place in the same voltage range as compared with the two-dimensional structure.

2.5 Results

The simulation results show that the spherical out-diffusion of the DNTUB is larger than expected because of the large NTUB depth. This depth is about 7.5 micron in the two-dimensional simulation. The spherical diffusion length is also in the same size from the top of the DNTUB finger to the direction of the two-dimensional case. In fact the two-dimensional situation is given when the DNTUB mask is enlarged by about 7 micron as compared to Fig. 4. This means that the DNTUB mask even can exceed the PTUB mask. However an enlargement of 7 microns would cause impact ionization near the surface's PTUB top. So the limiting case of the DNTUB enlargement is the dopant concentration of the two-dimensional simulation at the surface of the junction. This critical concentration is given when the DNTUB mask is shifted by 2 microns towards the PTUB mask (see Fig. 2.5).

The simulation results are validated by a set of test devices. Figure 5 shows the punch current dependence of the finger elongation starting with the initial layout (Fig. 4).

Another interesting effect is that the punch-through in the three-dimensional case does not occur directly under the symmetry line of the finger (see Fig. 7). The explanation is that the DNTUB dopands diffuse spherically while the PTUB dopands diffuse cylindrically coordinates. The punch current therefore has its maximum density near the edge of the PTUB mask.

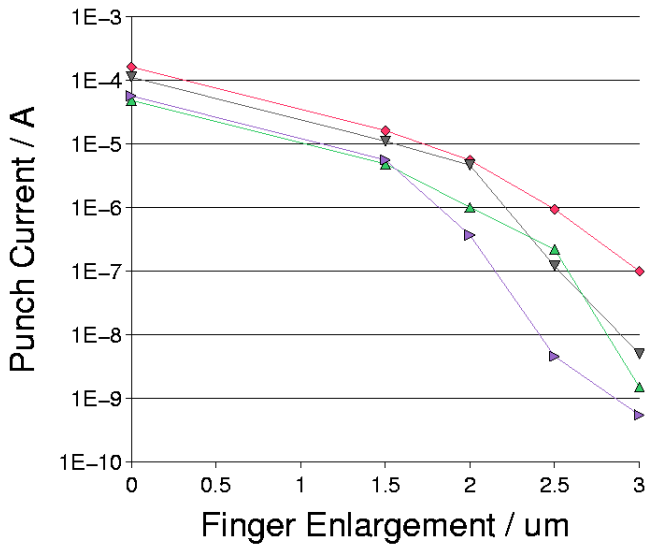

Figure 5: Measured punch current of four test devices, depending on the finger enlargement, substrate connected to -90V

With these careful considerations the device has been optimized to fulfill electrical strength, particularly with regard to punch-through between the junctions and breakdown by impact ionization. Without the outlined simulation methodology it would not have been possible to fully optimize the structure.

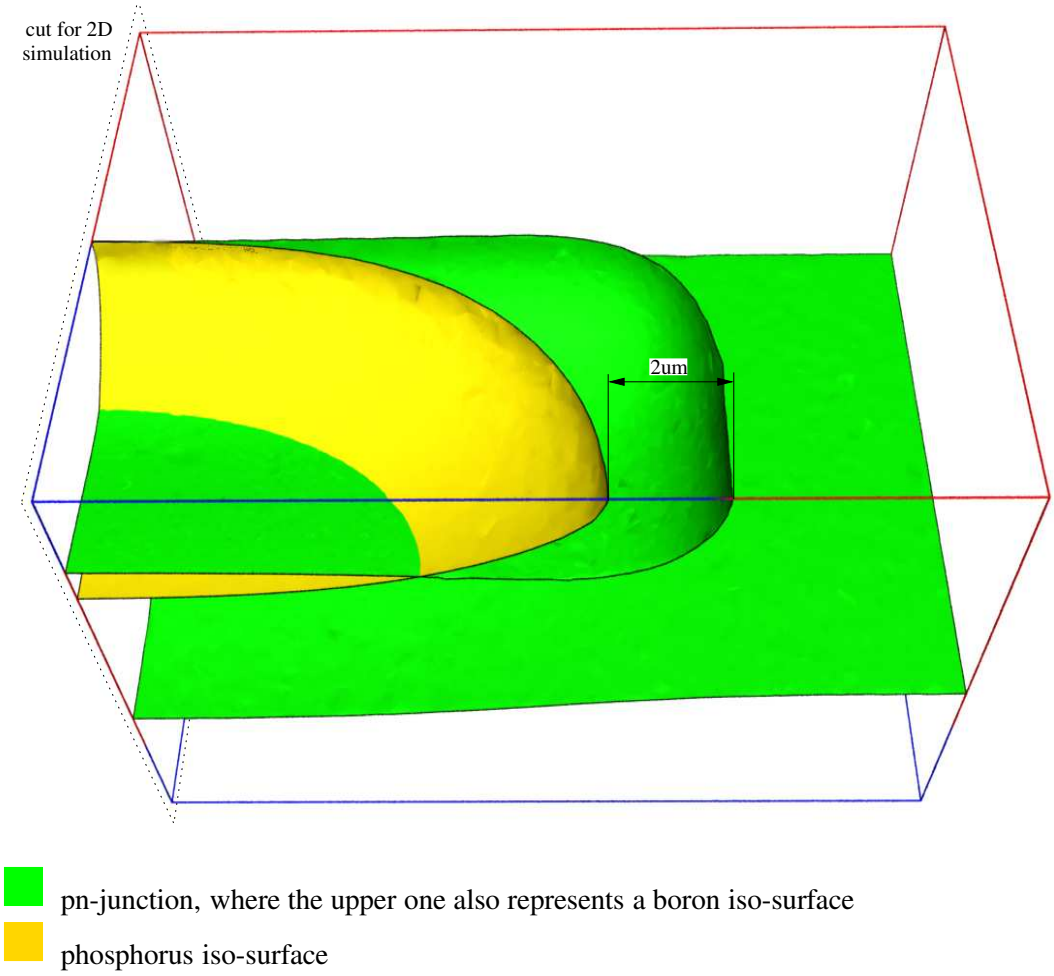

Figure 6: Relevant iso-surfaces of the junctions and the phosphorus doping with a simulation domain of 15.7μm x 10μm x 10μm

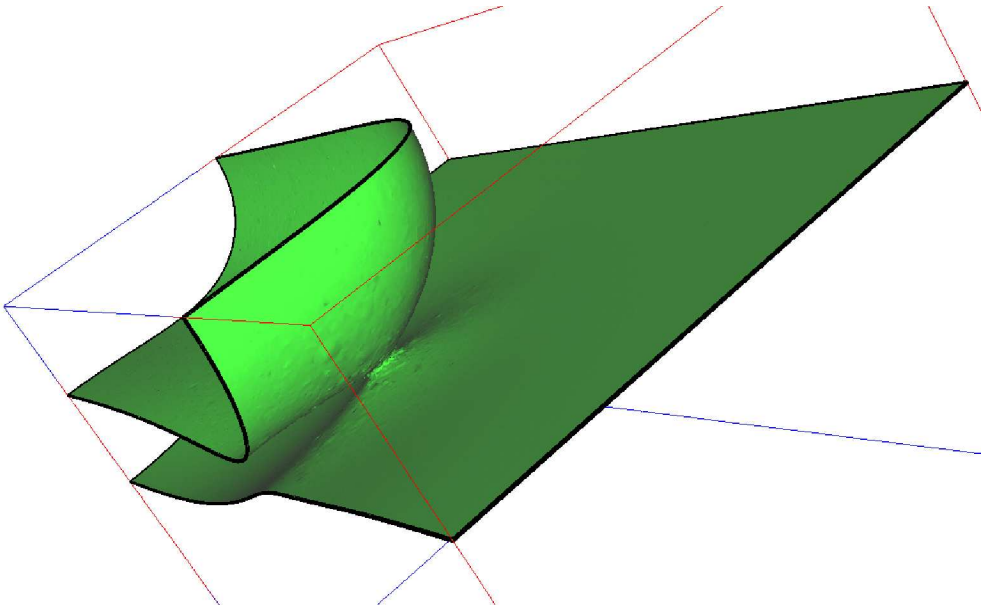

Figure 7: Surface surrounding the space charge region between both pn-junctions

3 TCAD Analysis of Gain Cell Retention Time for SRAM Applications

3.1 Introduction

To meet the demand for fast, nonvolatile memory, a further increase in the density of SRAM cells is inevitable. However, common SRAM technology still relies on the conventional six transistor cell, where downsizing is difficult. Recent papers present new approaches to increase the packing density of SRAM cells using advanced cell layouts.

An example is found in [16] where a MOS capacitance is used as storage node and two access transistors serve for independent read and write operations. The schematics of such a cell is shown in Fig. 8. The contacts are denoted by WWL (write word line), WBL (write bit line), RBL (read bit line) and RWL (read word line), respectively. While it is not a real SRAM circuit because of the volatility of the charge on the storage node, it offers the possibility of non-destructive read out due to capacitive coupling of the read transistor: when the storage node is charged, a positive voltage at the RWL contact suffices to open the read transistor, leading to a high current at the sense contact RBL. The sensing current is thus delivered by the RWL contact and does not reduce the charge on the storage node.

The cell can be fabricated with standard process steps and consumes much less die area as compared to a six transistor SRAM.

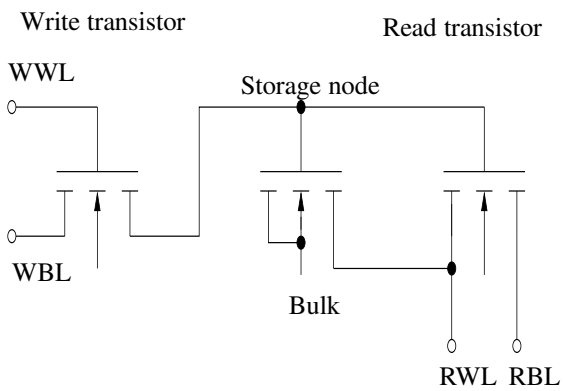

Figure 8: Schematic of the proposed gain cell.

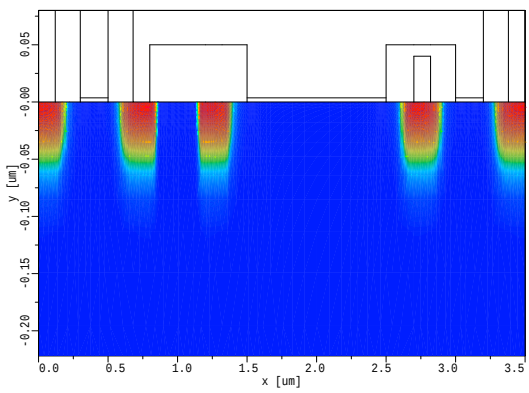

Figure 9: Doping profile of the memory cell.

For device simulation a complete three-dimensional doping profile of the cell is not necessary. Using standard process simulation tools, single MOSFET device profiles can be used for the creation of the doping profile of the whole cell. The two-dimensional doping profile of the simulated cell is shown in Fig. 9. The doping profile was produced using the doping profile of a standard 0.2μm gate length n-type MOSFET and the process simulation tool PROST2d. For such a memory cell two effects can be identified to affect the retention time: the leakage current through the access transistors and the gate tunneling current through the large gate area of the storage node.

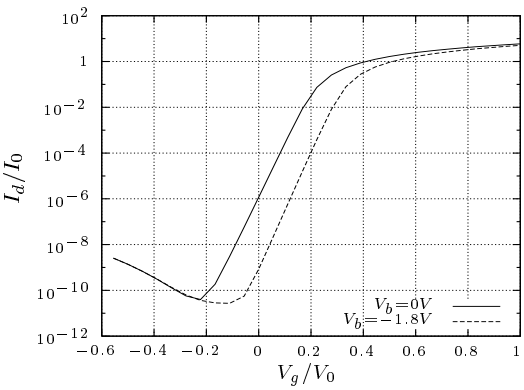

Figure 10: Transfer chracteristics of the NMOS device.

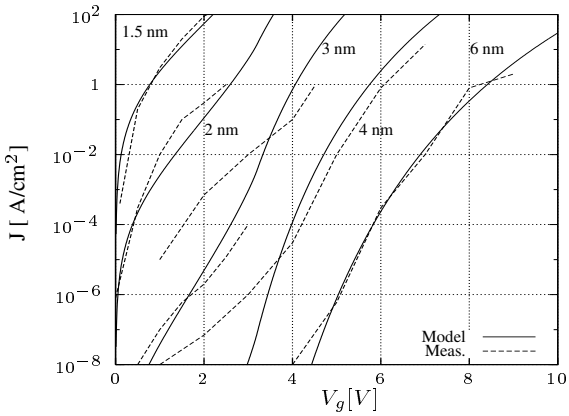

Figure 11: Gate Current density for different gate oxide thicknesses.

3.2 Inverse Modeling

We used the optimization tool SIESTA [17] and the device simulator MINIMOS-NT [18] to fit the simulation models for measured transfer characteristics at different bulk biases of the NMOS device. We chose the following parameters for inverse modeling: the bulk doping for accurate modeling of the bulk voltage induced shift of the threshold voltage, the band-to-band tunneling and Shockley-Read-Hall parameters to model the drain current increase for negative gate voltages, and the work function difference. The transfer characteristics is shown in Fig. 10 for different bulk voltages, the obtained agreement to measurement data is perfect. We tried several optimization schemes (genetic optimization, gradient based optimization and simulated annealing) and achieved the best fit using simulated annealing.

3.3 Modeling Tunneling Currents

Due to the large size of the MOS capacitor, the storage node is de-charged by gate tunneling currents. To give some order-of-magnitude estimations, we made use of a simple non-local, electric-field based tunneling model. For low gate voltages, the SiO_2 barrier is of trapezoidal shape. Using the standard WKB approximation, the tunneling current density evaluates as

$$J = \frac{q^3 m_{ox}}{8\pi h m_0 \Phi_b} \left(\frac{V_{ox}}{t_{ox}}\right)^2 \cdot \exp\left[-\frac{4 t_{ox}\sqrt{2 m_{ox}}}{3 q V_{ox}\hbar}\left(\Phi_b^{3/2} - (\Phi_b - qV_{ox})^{3/2}\right)\right] \tag{2}$$

where V_{ox} is the oxide voltage, t_{ox} the gate oxide thickness and Φ_b the barrier height. The other symbols have their usual meanings. The derivation of this equation can be found in [19], a similar equation is used in [20].

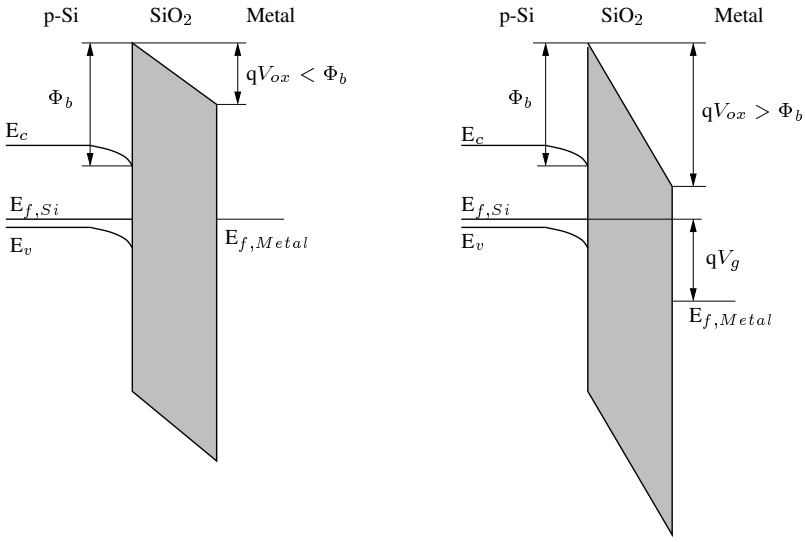

Figure 12: Tunneling barrier for the case of direct tunneling (left) and Fowler-Nordheim tunneling (right).

As shown in Fig. 12, (2) applies for $qV_{ox} < \Phi_b >$. Otherwise, the barrier is triangular, giving rise to Fowler-Nordheim tunneling causing a gate current density of

$$J = \frac{q^3 m_{ox}}{8\pi h m_0 \Phi_b} \left(\frac{V_{ox}}{t_{ox}}\right)^2 \cdot \exp\left(-\frac{4\sqrt{2m_{ox}}}{3qF_{ox}\hbar}\Phi_b^{3/2}\right) \tag{3}$$

which is the expression usually found in literature. We used literature data to calibrate the model, with the electron mass in the oxide m_{ox} as fitting parameter and Φ_b set to 3.1 eV. In the simulation we extracted the potential values at the oxide interfaces at equidistant lateral positions beneath the gate contacts. In Fig. 11 the tunneling current densities are plotted for an oxide mass of $m_{ox} = 0.47m_0$. It shows that the model, despite of its simplicity, is accurate enough to estimate the order of magnitude of the tunneling current density over a wide range of gate voltages. The measurement values were taken from [20], [21], and [22].

3.4 Contact Voltage Optimization

In addition to the gate tunneling current also the voltages at the contacts in the off-region have some influence on the retention time which is caused by the leakage current of the write transistor. The leakage current of the read transistor plays a minor role since only its gate is connected to the storage node. The write transistor leakage current shows no clear dependence on the contact voltages, as shown in Fig. 13. Also the transfer characteristics in Fig. 10 justifies the assumption that there exists a minimum of the leakage current. However, since there are five independent contact voltages (WWL, WBL, RWL, RBL and the bulk contact), optimization 'by hand' becomes difficult. Thus, we again used SIESTA for optimization, employing different optimization strategies. One constraint was that all voltages in the off-region had to stay below 0.5V. The default and optimum values for the contact voltages are shown in Tab. 2. For the read transistor, it turned out that positive voltages at the turned-off contacts can increase the retention time. Also, a positive bias on the bulk contact leads to a higher retention time.

	WWL	WBL	RWL	RBL	Bulk
Default	-0.2V	0V	0V	0V	-0.2V
Optimized	-0.398V	0.417V	0.249V	0.400V	0.417V

Table 2: Contact voltages

3.5 Transient Simulation

Finally, we show results of transient simulations using the gate current model and the optimized contact voltages. In Fig. 14 the cell de-charging curves are shown for the case of optimized and not optimized contact voltages and different gate oxide thicknesses. Only with gate oxides thinner than 2nm, a significant reduction in retention time can be seen. For optimized contact voltages, the retention time can be increased by nearly three orders of magnitude. This emphasizes the need for proper chosen contact voltages when using such devices.

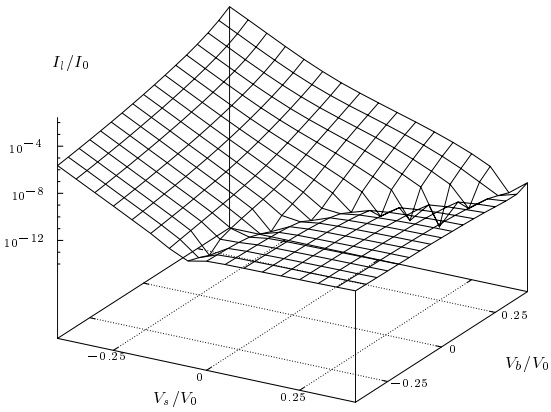

Figure 13: Effect of source and bulk voltage on the leakage current of the write transistor.

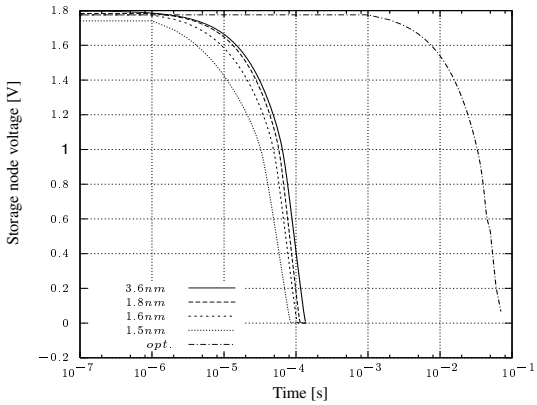

Figure 14: De-charging curves for different oxide thicknesses (left) and optimized curve (right).

3.6 Conclusion

We presented simulations of a new SRAM cell consisting of two transistors and one capacitor. We used inverse modeling and optimization techniques together with rigorous device simulation to analyze the retention time of the device. We showed that, even with a very simple gate current model, measured data can be fitted to some accuracy, valuable for first estimations. However, gate current induced charge loss is crucial only for gate oxides thinner than 2nm. Of higher importance is the right choice of the contact voltages, which can increase the retention time by orders of magnitude. Without TCAD based optimization it would have been cumbersome, if possible at all, to find the right optima.

4 A Qualitative Study on Global and Local Optimization Techniques for TCAD Analysis Tasks

4.1 Introduction

We compare the two well-known global optimization methods, simulated annealing and genetic optimization, to a local gradient-based optimization technique. We rate the applicability of each method in terms of the minimal achievable target value for a given number of simulation runs in an inverse modeling application.

The gradient-based optimizer used in the experiment is based on the Levenberg-Marquardt algorithm. The actual implementation (lmmin) was taken from MINPACK [23]. The genetic optimizer (genopt) is based on GALIB [24]. For the simulated annealing [25] optimizer (siman) an implementation by L. Ingber was taken. All optimizers are capable of evaluating several targets in parallel.

In our inverse modeling experiment the dopant concentration profile of an NMOS transistor should be identified. We use the deviation of computed $I_D V_D$ and $I_D V_G$ curves from measured ones as a target for optimization. The target function as delivered to the optimizer is determined by $\sqrt{(\vec{x} \cdot \vec{x})/N}$ where $\vec{x}$ is the N-dimensional error vector. The error vector is computed as a modified relative error: $100 \cdot (1 - I_c/I_m)$ for $I_c < I_m$ and $100 \cdot (I_m/I_c - 1)$ otherwise [26], where I_c and I_m denote the computed and measured currents, respectively. The dopant profiles are approximated by Pearson Type IV functions as described in [27]. Fig. 15

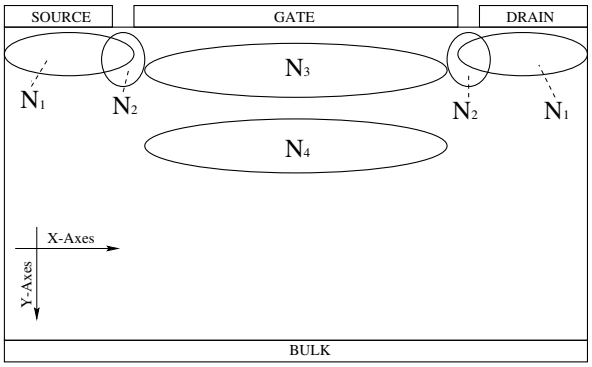

Figure 15: Two-dimensional device model with analytical doping peaks

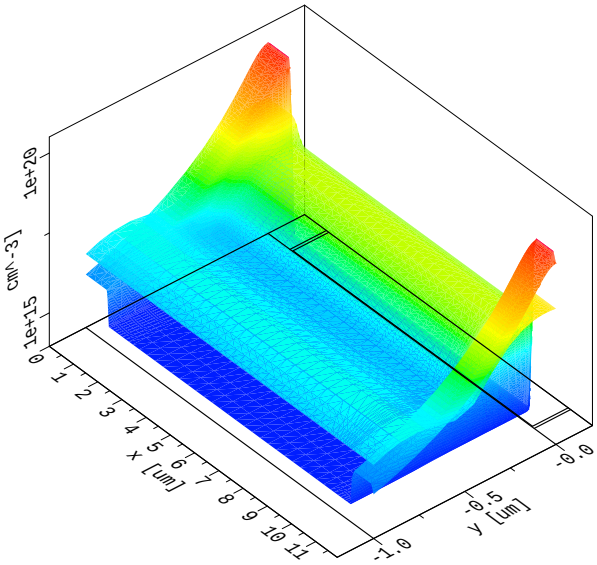

Figure 16: Plot of device with donors and acceptors

shows the two-dimensional model of the device under consideration. The elliptically shaped regions denote the analytical dopant concentrations. Fig. 16 shows a plot of the donor and acceptor concentrations and the geometry of a typical device.

A total of 27 free parameters was optimized. In order to utilize a cluster of workstations we used our simulation environment SIESTA [28, 29] to distribute the computational load. For the extraction of the curves the device simulator MINIMOS-NT [30, 31] was used.

4.2 Optimizers

4.2.1 Gradient-Based Optimizer

A gradient-based optimizer approximates the target function by a terminated Taylor series expansion:

$$f(\vec{x}_0 + \vec{x}) \approx f(\vec{x}_0) + (\nabla f(\vec{x}_0))^T \vec{x} + \frac{1}{2}\vec{x}^T \nabla^2 f(\vec{x}_0)\vec{x} \tag{4}$$

The actual optimization is performed iteratively. The direction and step width are determined by numerically computing the JACOBIAN and HESSIAN matrices of the target function. Our optimizer uses a finite-difference approximation of the first derivatives thus two evaluations for each parameter are necessary. The second derivatives are computed by using the gradient of the recent and of the last step and the HESSIAN of the last step (Broyden-Fletcher-Goldfarb-Shanno update [32]). The evaluations are independent from each other which means they can be carried out in parallel. The dependence of the number of evaluations on the number of free parameters limits the scalability of the optimizer and thus the utilization of the workstation cluster (for a small number of parameters).

The performance of the gradient-based methods strongly depends on the initial values supplied. Several optimization runs with different initial guesses might be necessary if no a priori knowledge (e.g., the result of a process simulation) about the dopant concentration profile is applied. Fig. 17

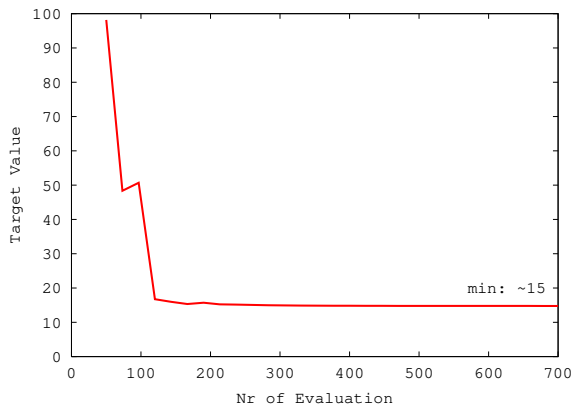

Figure 17: Progress of the gradient-based optimizer

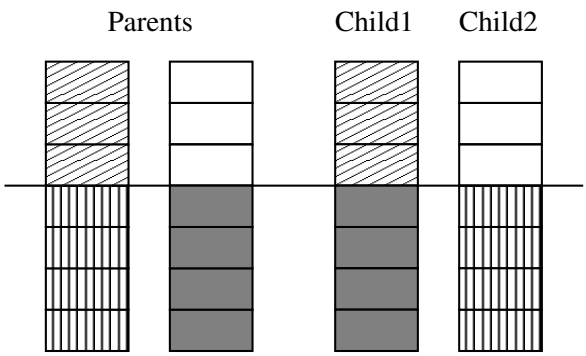

Figure 18: Crossover Operator

shows the evolution of the target values for a certain initial guess. In this example the optimizer was stopped at a local minimum. Care must be taken to provide physically sound bounds for all parameters to avoid simulation failures.

4.2.2 Genetic Algorithms

Genetic algorithms go back to [33]. A genetic algorithm (GA) is a so called population based search strategy. GA's maintain a set of points (genomes) in a function space. When the optimizer is started an initial population of genomes is chosen. The parameters of the genomes are initialized randomly but within given bounds. The fitness of the individuals in the population is then computed (in our case by means of a device simulation). The simulation result i.e. the target value is used for selecting individuals for reproduction. The library (GALIB) we used supports four different flavors of genetic algorithms namely

SIMPLE (as described in [34]), STEADY-STATE, INCREMENTAL and DEME. They differ in the way individuals are selected for mating, dying and for surviving. In case of the SIMPLE genetic algorithm the whole population is replaced each generation. The STEADY-STATE algorithm replaces only a part of the population. Some of the individuals survive into the next generation. The replacement percentage defines how many individuals are replaced. In the INCREMENTAL algorithm each generation consists of only one or two children. Finally, the DEME algorithm evolves several populations independently each with a STEADY-STATE algorithm. Each generation some individuals are migrated across the populations.

Genetic Reproduction

Reproduction is controlled by mutation and crossover operators. Crossover defines the procedure for generating a child from two parents. The crossover probability (P_{cross}) is used to decide whether the parents or their children are taken over into the next generation. Fig. 18 shows the one-point-crossover method, where a point is chosen randomly to determine which part of the genome to take from mother and father respectively. GALIB supports several crossover methods. For our experiments we used the one-point-crossover and two-point-crossover algorithms. For the optimization task crossover is the attempt to find better individuals by combining the parameters of the best individuals so far.

Mutation introduces new genetic material into a population. Mutation occurs with the probability P_{mut}. One parameter in a genome is replaced by a randomly chosen value (within the allowed range).

Genopt

Our genetic optimizer (genopt) is written using GALIB. For our application we obtained the best results with the STEADY-STATE algorithm. We used a replacement percentage of $P_{replace} = 0.7$ and a population size of 40. Since GALIB does not support parallel target evaluation our optimizer takes care of evaluating several jobs in parallel.

The parameters of genopt with the most impact are the crossover probability P_{cross} and the mutation probability P_{mut}. Several experiments with different crossover and mutation probabilities were carried out. Fig. 19 and Fig. 20 depict the evolution of the genetic algorithm for two different combinations of crossover and mutation probability and crossover method. The solid line is a plot of the best individual of each generation. Note that the best individual within a population sometimes occurs at a lower evaluation number thus appearing below the solid line.

The parameter combination depicted in Fig. 19 leads to the best result for our application.

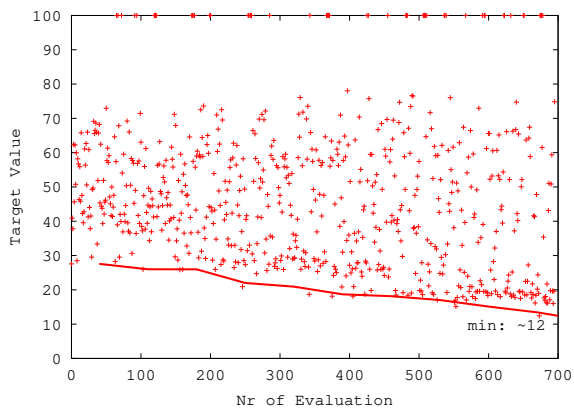

Figure 19: Evolution of the genetic optimizer for $P_{cross} = 0.9$, $P_{mut} = 0.2$ and two-point-crossover

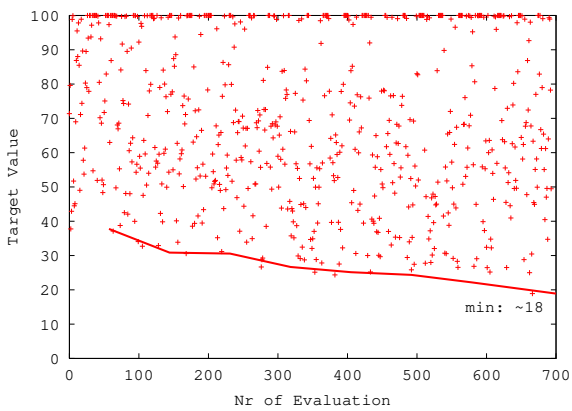

Figure 20: Evolution of the genetic optimizer for $P_{cross} = 0.8$, $P_{mut} = 0.3$ and one-point-crossover

4.2.3 Simulated Annealing

Simulated Annealing is an optimization technique which was first introduced by Kirkpatrick in 1983 [35]. It is comprised of three functional relationships: The generation function $g(\vec{x})$, where $\vec{x} = \{x^i; i = 1, D\}$ with dimension D, the acceptance function $h(\vec{x})$ and the annealing schedule function $T(k)$ with the time step k. The optimization itself takes place iteratively. Initially, the algorithm starts from a randomly chosen point from which the fitness is computed. Next a new point is chosen using $g(\vec{x})$. In case the fitness of this point is better than the fitness of the other one, the new point is taken over. In case the fitness is worse the point is accepted by a probability $h(\vec{x})$. Another point is always chosen based on the best point so far. With each iteration the probabilities for large deviations from the best point and for acceptance decrease. This results in a behavior where distant points are explored at the beginning (high temperature) but not generated or rejected respectively as the temperature cools down.

For the standard Boltzmann Annealing $g(\vec{x})$, $h(\vec{x})$ and $T(k)$ are given by:

$$g(\vec{x}) = (2\pi T)^{-\frac{D}{2}} \exp\left(-\frac{\Delta\vec{x}^2}{2T}\right), \tag{5}$$

$$h(\vec{x}) = \frac{1}{1 + \exp\left(\frac{E_{k+1} - E_k}{T}\right)}, \tag{6}$$

$$T(k) = \frac{T_0}{\ln k} \tag{7}$$

with the deviation $\Delta\vec{x} = \vec{x} - \vec{x}_0$ of the new state from the previous one. It was shown [36] that a global minimum will be found if the temperature is decreased no faster as given by (7).

Siman

Our simulator (siman) is based on the VERY FAST SIMULATED RE-ANNEALING [25] algorithm by L. Ingber. The algorithm defines a generation rate which allows for an exponentially decreasing time step function:

$$T_i(k) = T_{0i} \exp\left(-c_i k^{\frac{1}{D}}\right) \tag{8}$$

with $c_i = m_i \exp\left(-\frac{n_i}{D}\right)$, where m_i and n_i are tuning parameters. The values T_{0i} are the initial annealing temperatures.

To account for different sensibilities of the parameters the algorithm periodically re-scales the annealing time k. The range over which the more insensitive parameters are searched is stretched out with respect to the more sensitive parameters (RE-ANNEALING).

Fig. 21 shows the progress of siman. Standard parameter settings were used. Compared to genopt this optimizer reaches the same target value within approximately one third of evaluations.

4.3 Conclusion

We conclude that among the global optimization strategies we evaluated, simulated annealing seems to be best suited for the case of our inverse modeling application. We observed that for a larger number of

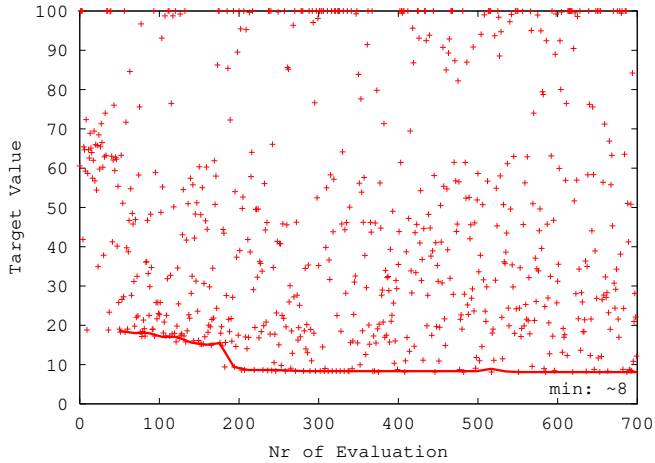

Figure 21: Evolution of the simulated annealing optimizer

evaluations (several thousands) siman delivered nearly optimal target values, whereas genopt's optima did not drop below a certain value. This calls for further experimenting with P_{cross} and P_{mut} and other parameters during the evolution. However, the optimal settings for these parameters are difficult to extract. We found that the VERY FAST SIMULATED RE-ANNEALING algorithm is faster than the STEADY-STATE genetic algorithm by at least a factor of three. This conforms to the experiments done by L. Ingber [37] who reports a speed difference of about one magnitude.

The local gradient-based method is the fastest if the initial guess is chosen appropriately but stops in a local minimum or even fails to converge. In this case the whole optimization must be restarted with a different initial guess.

Compared to a local optimizer the presented global optimization techniques demonstrate robust optimization strategies which are essential in cases where an appropriate initial guess is not available.

Further investigations will combine the advantages of global and local optimization techniques. One could imagine a scenario where for each globally found target value below a certain limit (e.g. 15), a separate local optimizer is tried for a certain time period. This combines the robustness of the global technique with the speed of the local one.

5 Optimization for TCAD Purposes Using Bernstein Polynomials

5.1 Introduction

The optimization of computationally expensive objective functions requires approximations that preserve the global properties of the function under investigation. The RSM approach of using multivariate polynomials of degree two can only preserve the local properties of a given function and is therefore not well-suited for global optimization tasks. In this paper we discuss generalized Bernstein polynomials that provide faithful approximations by converging uniformly to the given function. Apart from being useful for optimization tasks, they can also be used for solving design for manufacturability problems.

Automated TCAD optimization is difficult since the evaluation of the objective function is usually very computationally expensive. There are two main approaches: the first is to optimize the given objective function, and the second is to optimize an approximation of the objective function. Both approaches are implemented in the SIESTA(Simulation Environment for Semiconductor Technology Analysis) framework [17, 28]. The second approach relies on how good an approximation was chosen, and that it can be evaluated much faster than the original objective function so that conventional optimization algorithms requiring many more evaluations can be applied.

In the RSM (response surface methodology) [38] almost exclusively polynomials of degree two (or less) are used. This method, however, suffers from the fact that there is no reason why such an approximation should preserve the global properties of the given function: the set of of all polynomials of degree two or less is not dense in $C(X)$, $X \subset \mathbb{R}^p$ compact. Moreover, evaluating the objective function at more and more points does generally not improve the RSM approximation – these evaluations are wasted. A simple example for this fact are the functions $e_\lambda : x \mapsto \mathrm{e}^{\lambda x}$ which are ubiquitious in TCAD applications. Other examples are functions containing transitions from exponential to linear behavior.

Although the RSM approach can be improved by transforming the variables before fitting the polynomials, it has to be known a priori which transformations are useful and should be considered. If this knowledge is available, it can of course be applied to other optimization approaches as well.

To overcome the shortcoming of the RSM approach, we propose using generalized Bernstein polynomials for approximating objective functions.

We also note that a good approximation resembling the global properties of the objective function can be used for solving design for manufacturability problems. Furthermore, this method of computing approximations evidently gives rise to a recursive optimization algorithm. After a first approximation either further approximations of interesting areas are computed, or – if needed – the first approximation is refined using additional points.

5.2 Properties of Bernstein Polynomials

In this section we discuss some important properties of (generalized) Bernstein polynomials. In order to keep the formulas simple we will concern ourselves with functions defined on the (multidimensional) intervals $[0, 1] \times \cdots \times [0, 1]$. Using affine transformations it is straightforward to apply the results to arbitrary intervals.

The following theorem is due to Sergei N. Bernstein.

5.1 Theorem *Let $f : [0,1] \to \mathbb{R}$ be a continuous function. Then the Bernstein polynomials*

$$B_{f,n}(x) := \sum_{k=0}^{n} f\left(\frac{k}{n}\right)\binom{n}{k} x^k (1-x)^{n-k}$$

converge uniformly to f for $n \to \infty$.

A proof can be found in [39, p. 339]. If f even satisfies a Lipschitz condition, a stronger result can be shown giving an error bound.

5.2 Theorem *If f additionally satisfies a Lipschitz condition $|f(x)-f(y)| < L|x-y|$, then the inequality*

$$|B_{f,n}(x) - f(x)| < \frac{L}{2\sqrt{n}}$$

holds.

Additional to uniform convergence, also the derivatives of the approximation converge to those of the given function.

5.3 Theorem *If f has a continuous i-th order derivative $f^{(i)}(x)$ on $(0,1)$, then $B^{(i)}_{f,n}(x)$ converges uniformly to $f^{(i)}(x)$ on $(0,1)$.*

The proof for this theorem is still elementary but requires more careful analysis.

The generalization for a function of two variables is obtained by first approximating one variable and then the second. But using this straightforward method we can only prove pointwise convergence.

5.4 Theorem *Let $f : [0,1] \times [0,1] \to \mathbb{R}$ be a continuous function. Then the two-dimensional Bernstein polynomials*

$$B_{f,n}(x,y) := \sum_{k=0}^{n}\sum_{\ell=0}^{n} f\left(\frac{k}{n},\frac{\ell}{n}\right)\binom{n}{k}\binom{n}{\ell} x^k (1-x)^{n-k} y^\ell (1-y)^{n-\ell}$$

converge pointwise to f for $n \to \infty$.

This method can of course be applied recursively.

5.5 Theorem *Let $f : [0,1] \times \cdots \times [0,1] \to \mathbb{R}$ be a continuous function of m variables $x_1,\ldots,x_m$. Then the multi-dimensional Bernstein polynomials*

$$B_{f,n}(x_1,\ldots,x_n) := \sum_{k_1,\ldots,k_m=0}^{n} f\left(\frac{k_1}{n},\ldots,\frac{k_m}{n}\right)\prod_{j=1}^{m}\left(\binom{n}{k_j} x_j^{k_j}(1-x_j)^{n-k_j}\right)$$

converge pointwise to f for $n \to \infty$.

5.3 Examples

In this section we discuss two examples illustrating the properties of Bernstein polynomials, namely an analytical function and a two-dimensional inverse modeling example.

The example of the function $f : [0, 1] \times [0, 1] \to \mathbb{R}$,

$$f(x, y) := (1/2)\mathrm{e}^{-10((x-1/2)^2+(y-1/2)^2)} + \mathrm{e}^{-50((x-1)^2+(y-1)^2)}$$

shows that approximation using generalized Bernstein polynomials resembles the global properties of a given function better than using multivariate polynomials of degree 2 or less, even when using a small number of lattice points. The two approaches are compared in Fig. 22. In the left hand figure, f is plotted at the $11 \cdot 11$ lattice points that were used for calculating the two-dimensional Bernstein polynomial $B_{f,10}(x, y)$ and the least squares fit $rsm(x, y)$ of degree 2. f and $B_{f,10}$ have two local maxima on $[0, 1] \times [0, 1]$, whereas rsm has only one. Their respective values are (up to six digits): $f(0.5, 0.5) = 0.5$, $f(0.999661, 0.999661) = 1.00338$; $B_{f,10}(0.500674, 0.500674) = 0.331634$, $B_{f,10}(1, 1) = 1.00337$; $rsm(0.696706, 0.696706) = 0.283076$.

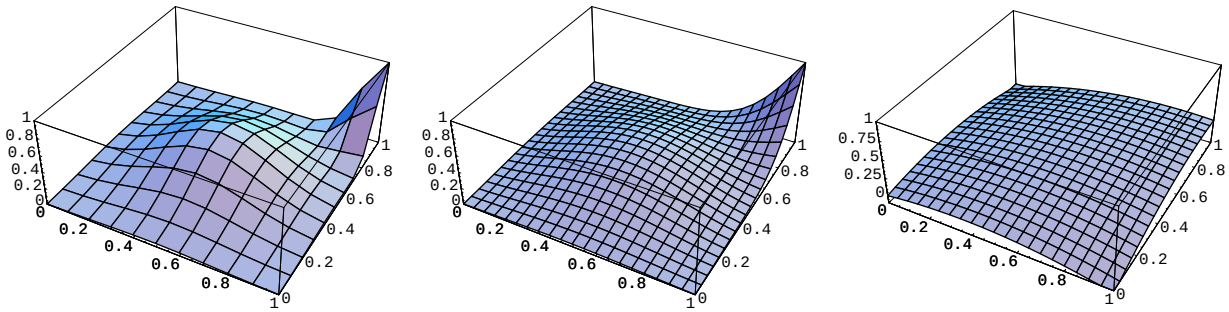

Figure 22: Comparison of $11 \cdot 11$ lattice points of f (left), the Bernstein approximation $B_{f,10}$ (middle, the variables have been scaled to the interval $[0, 1]$), and the RSM approximation rsm (right) as found by MATHEMATICA's `Fit` function.

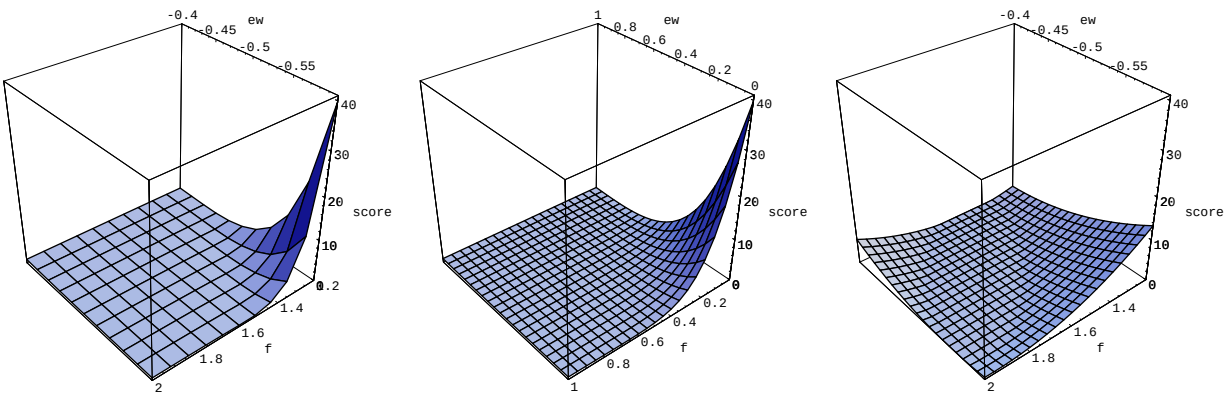

Figure 23: Comparison of the computed lattice points (left), the Bernstein approximation (middle), and the RSM approximation (right) as found by MATHEMATICA's `Fit` function.

The second, real world example stems from minimizing the leakage current of a novel SRAM storage cell [40]. First, we extracted seven parameters from the drain currents of the select transistor of the storage cell and tried to fit two transfer characteristics (two bulk voltages, two times 27 points). The seven variables

were `ew`, the work function of the gate material, `sr`, the source resistance, `f`, a parameter controlling the doping, and four variables pertaining to the Shockley–Read–Hall model [31, page 71]. In the second step the extracted values were used when minimizing the leakage current.

In the course of the inverse modeling task it was found that two variables, namely the parameter of the gate material (`ew`) and the parameter controlling the doping (`f`), have a major influence on the result. For further investigations, these remaining variables were then fixed at the values of the minimum found, and the objective function was evaluated at $11 \cdot 11$ lattice points with these two most sensitive parameters (cf. Fig. 23, left). Using these points, two approximations were calculated: the two-dimensional Bernstein polynomial (where the variables were scaled to the interval $[0, 1]$), and the least squares approximation from the set of all polynomials of degree two or less (cf. Fig. 23). Again the RSM approximation is misleading.

5.4 Conclusion

For optimization tasks involving computationally expensive functions, we propose using multivariate Bernstein polynomials for approximating objective functions instead of the conventional RSM approach of using polynomials of degree two or less. We show that this approach is mathematically sound and present two examples illustrating its advantages.

References

[1] F. W. Grover. *Inductance Calculations: Working Formulas and Tables*. D. Van Nostrand Company, New York, 1946.

[2] C. Harlander, R. Sabelka, and S. Selberherr. Inductance Calculation in Interconnect Structures. In *Proc. 3rd Intl. Conf. on Modeling and Simulation of Microsystems*, pp 416–419, San Diego, California, USA, 2000.

[3] A. H. Stroud. *Approximate Calculation of Multiple Integrals*. Prentice-Hall, Englewood Cliffs, N.J., 1971.

[4] R. Sabelka and S. Selberherr. A finite element simulator for three-dimensional analysis of interconnect structures. *Microelectronics Journal*, 32(2):163–171, 2001.

[5] W. Pyka, R. Martins, and S. Selberherr. Optimized Algorithms for Three-Dimensional Cellular Topography Simulation. *IEEE J.Technology Computer Aided Design*, 2000. `http://www.ieee.org/products/online/journal/tcad/accepted/Pyka-March00/`.

[6] R. Martins, W. Pyka, R. Sabelka, and S. Selberherr. Modeling Integrated Circuit Interconnections. In *Proc. Intl. Conf. on Microelectronics and Packaging*, pp 144–151, Curitiba, Brazil, 1998.

[7] R. Martins and S. Selberherr. Layout Data in TCAD Frameworks. In *Modelling and Simulation*, pp 1122–1126. Society for Computer Simulation International, 1996.

[8] P. Fleischmann, W. Pyka, and S. Selberherr. Mesh Generation for Application in Technology CAD. *IEICE Trans.Electron.*, E82-C(6):937–947, 1999.

[9] Robert Bauer and Siegfried Selberherr. Preconditioned CG-Solvers and Finite Element Grids. In *Proc. CCIM*, volume 2, Breckenridge, USA, 1994.

[10] W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Prentice-Hall, 1996.

[11] G. Leonhardt and W. Fichtner. Acceleration of Inductance Extraction by Means of the Monte Carlo Method. Technical Report 99/8, Integrated Systems Laboratory, ETH Zürich, 1999.

[12] A. Hössinger, S. Selberherr, M. Kimura, I. Nomachi, and S. Kusanagi. Three-Dimensional Monte-Carlo Ion Implantation Simulation for Molecular Ions. In *Electrochemical Society Proceedings*, volume 99-2, pp 18–25, 1999.

[13] M. Radi, E. Leitner, E. Hollensteiner, and S. Selberherr. Analytical Partial Differential Equation Modeling Using AMIGOS. In *Proc. IASTED International Conference Artificial Intelligence and Soft Computing*, pp 423–426, Banff, Canada, 1997.

[14] P. Fleischmann, R. Sabelka, A. Stach, R. Strasser, and S. Selberherr. Grid Generation for Three-Dimensional Process and Device Simulation. In *Simulation of Semiconductor Processes and Devices*, pp 161–166, Tokyo, Japan, 1996. Business Center for Academic Societies Japan.

[15] H.J. Dirschmid. *Einführung in die mathematischen Methoden der theoretischen Physik*. Vieweg, 1976.

[16] N. Ikeda, T.Terano, H. Moriya, T. Emori, and T. Kobayashi. A Novel Logic Compatible Gain Cell with two Transistors and one Capacitor. In *Symposium on VLSI Technology Digest of Technical Papers*, pp 168–169, 2000.

[17] C. Heitzinger and S. Selberherr. An Extensible TCAD Optimization Framework Combining Gradient Based and Genetic Optimizers. In *Proc. SPIE International Symposium on Microelectronics and Assembly: Design, Modeling, and Simulation in Microelectronics*, pp 279–289, Singapore, 2000.

[18] T. Simlinger, H. Kosina, M. Rottinger, and S. Selberherr. MINIMOS-NT: A Generic Simulator for Complex Semiconductor Devices. In H.C. de Graaff and H. van Kranenburg, editors, *25th European Solid State Device Research Conference*, pp 83–86, Gif-sur-Yvette Cedex, France, 1995. Editions Frontieres.

[19] J. P. Shiely. *Simulation of Tunneling in MOS devices*. Dissertation, Duke University, 1999.

[20] K. F. Schuegraf, C. C. King, and C. Hu. Ultra-thin Silicon Dioxide Leakage Current and Scaling Limit. In *Symposium on VLSI Technology Digest of Technical Papers*, pp 18–19, 1992.

[21] N. Yang, W. K. Henson, J. R. Hauser, and J. J. Wortman. Modeling Study of Ultrathin Gate Oxides Using Direct Tunneling Current and Capacitance-Voltage Measurements in MOS Devices. *IEEE Trans.Electron Devices*, 46(7), 1999.

[22] S. H. Lo, D. A. Buchanan, Y. Taur, and W. Wang. Quantum-Mechanical Modeling of Electron Tunneling Current from the Inversion Layer of Ultra-Thin-Oxide nMOSFETs. *IEEE Trans.Electron Devices*, 18(5), 1997.

[23] J. J. Moré, D. C. Sorensen, K. E. Hillstrom, and B. S . Garbow. *The MINPACK Project,*. Sources and Development of Mathematical Software. Prentice-Hall, Englewood Clifs, NJ, 1984.

[24] M. Wall. GAlib A C++ Library of Genetic Algorithm Components. *Massachusetts Institute of Technology*, 2000. `http://lancet.mit.edu/ga`.

[25] L. Ingber. Very Fast Simulated Re-Annealing. *Mathematical Computer Modelling*, 12:967–973, 1989. `http://www.ingber.com/asa89_vfsr.ps.gz`.

[26] R. Plasun. *Optimization of VLSI Semiconductor Devices*. Dissertation, Technische Universität Wien, 1999. `http://www.iue.tuwien.ac.at/diss/plasun/diss-new/diss.html`.

[27] S. Selberherr. *Analysis and Simulation of Semiconductor Devices*. Springer, Wien, New York, 1984.

[28] R. Strasser, R. Plasun, and S. Selberherr. Practical Inverse Modeling with SIESTA. In *Simulation of Semiconductor Processes and Devices*, pp 91–94, Kyoto, Japan, 1999.

[29] C. Heitzinger and S. Selberherr. An Extensible TCAD Optimization Framework Combining Gradient Based and Genetic Optimizers. *in Proc. International Symposium on Microelectronics and Assembly, Singapore 2000*, pp 279–289, 2000.

[30] T. Grasser, V. Palankovski, G. Schrom, and S. Selberherr. Hydrodynamic Mixed-Mode Simulation. In K. De Meyer and S. Biesemans, editors, *Simulation of Semiconductor Processes and Devices*, pp 247–250. Springer, Leuven, Belgium, 1998.

[31] T. Binder, K. Dragosits, T. Grasser, R. Klima, M. Knaipp, H. Kosina, R. Mlekus, V. Palankovski, M. Rottinger, G. Schrom, S. Selberherr, and M. Stockinger. *MINIMOS-NT User's Guide*. Institut für Mikroelektronik, 1998.

[32] W.T. Vetterling and S.A. Teukolsky. *Numerical Recipes*. Cambridge University Press, 1986.

[33] J. Holland. Adaption in Natural and Artificial Systems. *University of Michigan Press, Ann Arbor, MI*, 1975.

[34] D. E. Goldberg. *Genetic Algorithms in Search and Optimization*. Addison-Wesley Pub. Co., 1989.

[35] S. Kirkpatrick, C.D. Gelatt Jr, and M.P. Vecchi. Optimization by Simulated Annealing. *Science*, 220, no. 4598:671–680, 1983.

[36] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distribution and the Bayesian Restoration in Images. *IEEE Trans. Patt. Anal. Mac. Int.*, 6:721–741, 1984.

[37] L. Ingber. Genetic Algorithms and Very Fast Simulated Re-Annealing: A Comparision. *Mathematical and Computer Modelling*, 16:87–100, 1992. `http://www.ingber.com/asa92_saga.ps.gz`.

[38] G.E.P. Box and N.R. Draper. *Empirical Model-Building and Response Surfaces*. Wiley, New York, 1987.

[39] I.S. Berezin and N.P. Zhidkov. *Computing Methods*, volume 1. Pergamon Press, 1965.

[40] N. Ikeda, T.Terano, H. Moriya, T. Emori, and T. Kobayashi. A Novel Logic Compatible Gain Cell with two Transistors and one Capacitor. In *Symposium on VLSI Technology Digest of Technical Papers*, pp 168–169, 2000.