VISTA Status Report December 2006

R. Entner, R. Heinzl, A. Nentchev,
E. Ungersböck, M. Wagner, S. Selberherr

Institute for Microelectronics
Technical University Vienna
Gußhausstraße 27-29
A-1040 Wien, Austria

Contents

1 Negative Bias Temperature Instability Modeling for High-Voltage Oxides at Different Stress Temperatures **1**

1.1 Introduction . . . 1

1.2 The Reaction-Diffusion Model . . . 1

1.3 Dispersive Hydrogen Transport . . . 1

1.4 TCAD Simulation . . . 2

1.5 Experiment and Results . . . 2

1.6 Conclusion . . . 2

2 A Generic Topology Library **4**

2.1 Introduction . . . 4

2.2 Motivation . . . 4

2.3 Formal Specification . . . 5

2.4 Generic Topology Library . . . 7

2.4.1 Topological Properties of Data Structures . . . 8

2.4.2 Finite Cell Complexes . . . 8

2.5 Data Access . . . 11

2.6 GTL Architecture . . . 11

2.7 Outlook . . . 12

2.8 Conclusion . . . 13

3 Heatring - Smart Investigation of Temperature Impact on Integrated Circuit Devices **14**

3.1 Motivation . . . 14

3.2 Investigated Structure . . . 14

3.3 Thermal Simulation and Measured Results of the Heatring SLM . . . 14

3.4 Theoretical Background . . . 15

3.4.1 Electro-Thermal Simulation . . . 15

3.4.2 Boundary Conditions . . . 17

3.4.3 Initial Conditions . . . 17

3.5 Conclusion . . . 17

4 Monte Carlo Study of Electron Transport in Strained Silicon Inversion Layers 18

4.1 Introduction 18

4.2 Inclusion of the Pauli Principle in Monte Carlo Simulations 18

4.2.1 Comparison of Algorithms 19

4.3 Results 19

4.4 Conclusion 20

4.5 Acknowledgement 21

5 Quantum Correction for DG MOSFETs 22

5.1 Introduction 22

5.2 Approach 22

5.3 Results 24

5.4 Conclusion 25

5.5 Acknowledgement 25

1 Negative Bias Temperature Instability Modeling for High-Voltage Oxides at Different Stress Temperatures

The temperature bias instability of high-voltage oxides is analyzed. For the investigation of negative bias temperature instability (NBTI) we present an enhanced reaction-diffusion model including trap-controlled transport, the amphoteric nature of the P_b centers at the Si/SiO_2 interface, Fermi-level dependent interface charges, and fully self-consistent coupling to the semiconductor device equations. Comparison to measurement data for a stress/relaxation cycle and a wide range of temperatures shows excellent agreement.

1.1 Introduction

Negative bias temperature instability (NBTI) has come to the forefront of scientific interest. It occurs in p-type MOS devices stressed with negative gate bias at elevated temperatures. In particular for thicker oxides, as used in high-voltage devices, evidence has been found that the degradation is dependent on two major effects, the generation of interface traps N_it at the Si/SiO_2 interface and the transport of a mobile, hydrogen related species into the dielectric. The generation of interface traps leads to a shift of important transistor parameters such as the threshold voltage V_th, the drain current I_d, the transconductance g_m, and the off current I_off. Due to the need for accurate prediction of device and circuit lifetimes, modeling and simulation of the degradation physics has gained importance. There are two important factors for accurate modeling: (a) the physics of the degradation mechanisms have to be modeled as precisely as possible (b) the experimental and measurement setup must lead to an exact description of the device state. Here, especially the applied measurement technique needs special attention, as the method used for evaluating NBTI degradation can have a considerable impact on life-time extrapolation results [1, 2].

1.2 The Reaction-Diffusion Model

The original reaction-diffusion (RD) model was proposed by Jeppson and Svensson thirty years ago [3]. Since then the model has been continuously refined. It describes the degradation process as a reaction at the Si/SiO_2 interface generating an interface state, N_it, as well as releasing a mobile hydrogen related species, N_X. This generation process is described as

$$\frac{\partial N_\mathrm{it}(t)}{\partial t} = \underbrace{k_\mathrm{f}(N_0 - N_\mathrm{it}(t))}_{\text{generation}} - \underbrace{k_\mathrm{r} N_\mathrm{it}(t) N_X(0,t)^{1/a}}_{\text{annealing}} . \tag{1}$$

where k_f is the interface-trap generation and k_r the annealing rate. The symbol N_0 denotes the initial number of electrically inactive Si-H bonds and $N_X(0,t)$ is the surface concentration of the diffusing species. The value of a gives the order of the reaction. Originally, neutral hydrogen, H^0, was proposed which is obtained with $a = 1$. For molecular hydrogen, H_2, $a = 2$ where the molecule is assumed to be formed in the vicinity of the interface

$$\mathrm{SiH} + \mathrm{h}^+ \rightleftharpoons \mathrm{Si}^+ + \tfrac{1}{2}\mathrm{H}_2 . \tag{2}$$

The equilibrium of the forward and backward reaction is controlled by the hydrogen density at the interface $N_X(0,t)$. Thus, the transport mechanism of the hydrogen species away from the interface characterizes the degradation mechanism, controlling for example the V_th shift. The original reaction-diffusion model describes the transport as a purely diffusive mechanism which is expressed by the regular diffusion equation

$$\frac{\partial N_X(x,t)}{\partial t} = D\,\nabla^2 N_X(x,t) . \tag{3}$$

Here D is the diffusivity of the hydrogen species in the dielectric. As boundary condition for the Si/SiO_2 interface the influx of the newly created species has to be considered as $\partial N_\mathrm{it}/\partial t/a$.

1.3 Dispersive Hydrogen Transport

Instead of using the standard diffusion equation [3, 4] we assume trap-controlled, dispersive, transport of the hydrogen species [1]. The species N_X consists of conducting, N_c, and trapped, N_t, particles. The trapped particles are distributed in energy where the density at a trap energy-level E_t is given as $\rho(\mathbf{x}, E_\mathrm{t}, t)$. The trapped particles do not contribute to the transport. Thus, (3) transforms to

$$\frac{\partial N_\mathrm{c}(\mathbf{x},t)}{\partial t} + \frac{\partial N_\mathrm{t}(\mathbf{x},t)}{\partial t} = D\nabla^2 N_\mathrm{c}(\mathbf{x},t) . \tag{4}$$

At each trap energy-level a rate equation describes the dynamics between trapping and de-trapping as

$$\frac{\partial \rho(E_\mathrm{t})}{\partial t} = c(E_\mathrm{t}) N_\mathrm{c}\Big(g(E_\mathrm{t}) - \rho(E_\mathrm{t})\Big) - r(E_\mathrm{t})\rho(E_\mathrm{t}) . \tag{5}$$

Here, $c(E_\mathrm{t})$ and $r(E_\mathrm{t})$ are the energy-dependent capture and release rates, and $g(E_\mathrm{t})$ is the trap density-of-states (DOS), where commonly an exponential distribution is assumed.

1.4 TCAD Simulation

The reaction-diffusion model has been implemented in a two- and three-dimensional numerical device simulator [5] and the model equations are coupled fully self-consistently to the semiconductor device equations. The benefits are manifold, since now the local oxide field, hole concentration, charged carriers, and fast interface states (Fermi-level dependent charges) can be included in both the device equations and the NBTI model. Due to the availability of the solution of the semiconductor equations, the trap generation rate can be expressed as a function of the surface hole concentration p_s and the oxide electric field E_ox as $k_\mathrm{f} = k_{\mathrm{f},0}\, p_\mathrm{s}/p_\mathrm{ref}\, \exp(E_\mathrm{ox}/E_\mathrm{ref})$ [6] instead of using estimates for p and a constant E_ox. The symbols $k_{\mathrm{f},0}$, p_ref, and E_ref denote the reference values for the generation rate, the hole concentration and the electric field.

The amphoteric nature of the P_b centers at the Si/SiO_2 interface, which form the interface-traps N_it, is considered. The DOS of the P_b centers in the Si bandgap forms two distinct peaks of Gaussian shape [7]. P_b centers in the upper peak are assumed acceptor-like. They are electrically neutral when the Fermi-level is below the trap, and negatively charged otherwise. The traps in the lower peak are donor-like, neutral when the Fermi-level is energetically above and positively charged otherwise. The consideration of the exact DOS is important when the Fermi-level is not close to the valence band, as in measurement cycles.

For the comparison of measurements to simulation results the complete dynamics of degradation and annealing during the measurement intervals have to be taken into account. In our simulations we mimic the whole measurement procedure as closely as possible to reflect the real world conditions and to reduce the error due to the measurement delays.

1.5 Experiment and Results

The device under test was a 48 nm high-voltage oxide stressed for 1000 seconds at $V_\mathrm{g} = -25\,\mathrm{V}$ at four different temperatures (100, 125, 150, and 175 °C). A 1000 seconds relaxation phase with $V_\mathrm{g} = 0\,\mathrm{V}$ followed the stress phase. To determine the threshold voltage shift ΔV_th, the stress was interrupted for two seconds at each measurement point to perform a gate voltage sweep from $0\,\mathrm{V}$ to $-2\,\mathrm{V}$. During this period a remarkable amount of relaxation can be observed. Thus, it is crucial to include the recovery process in the model. The important advantage is that the threshold voltage can now be extracted from the simulation results in the same way as in the measurements. In contrast to the standard reaction-diffusion model (Figure 1), the extended model (Figure 2) shows excellent agreement with measurement data for a wide range of temperatures, which can be achieved using a single set of model parameters. During the faster process within the first few seconds the annealing is attributed to re-passivation of dangling Si/SiO_2 interface bonds with hydrogen from shallow traps close to the interface. After the consumption of all quickly available hydrogen, additional hydrogen can only be provided by de-trapping from deep traps in the oxide bulk, which is a slower process. As the transition is seamless, there is no change in the time exponent during annealing. To properly capture the relaxation phase, the DOS used in the dispersive transport equation is modeled by an exponential tail for shallow traps for fast trapping and de-trapping, while the slow process is governed by deep traps given by an additional Gaussian peak well below the hydrogen conduction band.

1.6 Conclusion

We have presented an enhanced NBTI reaction-diffusion model with dispersive transport and fully self-consistent coupling to the semiconductor device equations. The amphoteric nature of the P_b centers and its Fermi-level dependent charge state is modeled. The implementation of the model in a multi-dimensional numerical device simulator allows us to directly use many commonly approximated quantities such as the oxide electric field or the interface hole concentration in a self-consistent manner. The model has been calibrated to measurement data of a high-voltage MOSFET structure at a wide range of temperatures for both, a stress and a relaxation cycle. Here the full measurement setup has been taken into account and very good agreement has been shown. To extend the model from thick SiO_2 dielectrics to thin, nitrided oxides of state-of-the-art logic MOSFETs it might be necessary to include additional effects such as hole trapping.

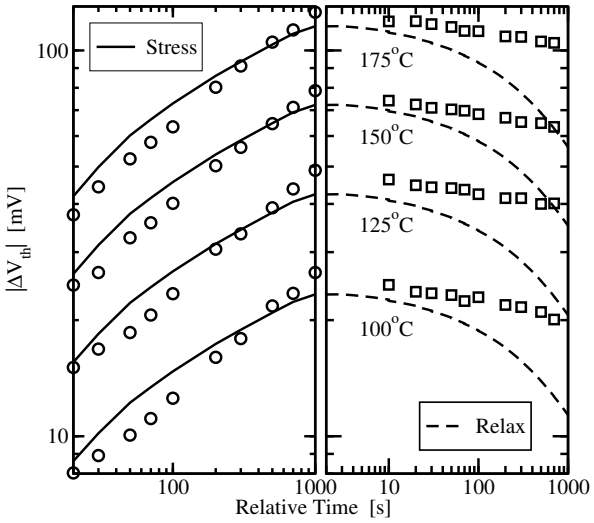

Figure 1: Comparison of the measurements to simulation results using the standard reaction-diffusion model. The model fails to reproduce measurement data, especially in the relaxation phase.

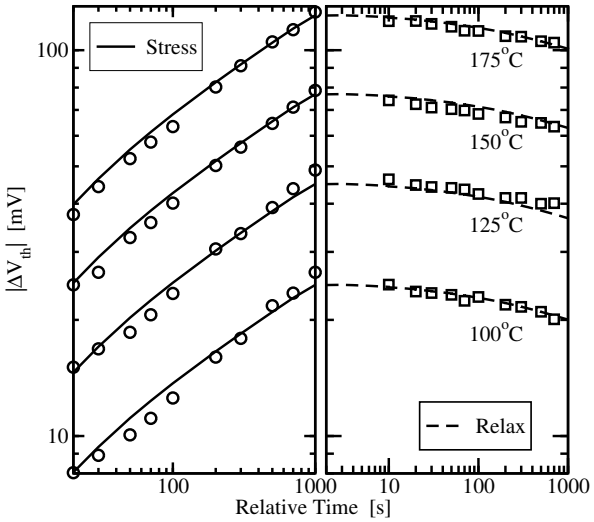

Figure 2: Comparison of the measurements to simulation results using the extended model. The slope of $n = 0.31$ during the stress phase is very well matched with the trap-controlled transport model and the agreement in the relaxation phase is excellent.

2 A Generic Topology Library

We present a generic topology library that is based on topological space and combinatorial properties. A notation is introduced whereby data structures can be described by their topological cell dimensions and internal combinatorial properties. A common interface for different types of data structures is presented. Various issues of iteration of these data structures can be explained from the topological properties. Using this multi-dimensional topology library we introduce new possibilities for functional programming in the field of scientific computing.

2.1 Introduction

In this work we investigate internal topological and combinatorial properties of data structures and the effect on their interfaces. Generic interfaces to data structures have proven to be highly successful means of generic programming. With the great achievement of accessing all data structures in a minimal but concise way, generic programming has emerged. A detailed analysis of generic programming is given in [8], where this topic is introduced from a theoretical point of view, namely category theory. A lot of insight is gained through this approach and a solid base has been achieved with this theory. Our work deals with the basic nature of topological spaces related only to data structures and is based on GrAL [9]. This is not as general as the category theory approach, but the basic features and issues are exposed.

Usually programmers have to know the **specific properties** of data structures to achieve the best performance of an algorithm. A simple example is the iteration and data access within a `std::vector`, which is constant, whereas the insertion or deletion uses linear time. This is relevant for the actual run-time behavior of all implemented algorithm applied to it. Closely related to this issue is the fact that the C++ STL algorithms use the most basic iteration mechanism for the access to data structures, the `forward` iterator mechanism. The optimal way of iteration of containers can often not be achieved, because linear iteration is simply not optimal [8], such as traversing a `std::map` or higher-dimensional topological structures, e.g., `boost::graph` from the Boost Graph library [10]. We introduce (Section 2.4.1) a unified data structure definition, where only the dimension and the combinatorial properties of topological spaces are specified. This can also be accomplished automatically at compile-time, based on requirements of algorithms.

Modern application design requires the utilization of data structures in several dimensions. Especially the field of scientific computing uses different topological elements to discretize partial differential equations (PDE). Various approaches are available such as the STL containers, the BGL, and for grids the GrAL [9]. However, a **standardized interface** to these data structures is missing. We introduce a basic interface (Section 2.4.2) for different dimensions of data structures based on topological and combinatorial properties.

A major issue of generic programming is the treatment of **data structure iteration and data access** [11], but the upcoming C++0x standard does not yet include this insight [12]. Therefore we use the property map concept [11], which is presented in Section 2.5 to utilize an extra data space. Briefly, the combination of iteration and access leads to a miscategorized algorithm specialization.

Our search for a general data structure library for the needs of scientific computing has shown that the topological structures of different STL containers and BGL mechanisms can be abstracted and generalized to a multi-dimensional generic topology library (GTL). We do not only separate the data access and iteration [11], but also provide a formal description of the underlying topological space with emphasis on the combinatorial properties:

topological space + data type = data structure

With a formalization of the topological properties and the iteration mechanism this approach renders a new possibility of the functional programming paradigm (Section 2.7) which is emerging in C++ [13, 14]. Up to now, functional expressions lack the support of a unique interface for all different kinds of data structure iteration. As we present in a generic discretization library for the discretization of various partial differential equations (GDL [15]), the full power of functional programming is revealed with consistent topological data structure. Note, the GTL is not restricted to applications for scientific computing, simple iterations can be specified elegantly as well.

2.2 Motivation

Our motivation for developing generic libraries is derived from the need in high performance applications in the field of scientific computing, especially in Technology Computer Aided Design (TCAD). Briefly, TCAD deals with the assembly of large equation systems by utilizing discretized partial differential equations from different fields of physics. All types of PDEs (elliptic, parabolic, hyperbolic) have to be considered for the various types of problems from the fields of semiconductor simulation [16]. Different grid types and dimensions of

topological elements, linear and nonlinear solvers with their associated numerical issues have to be considered during application development and demand great care to ensure high software quality while also addressing performance issues.

Our institute has a long history in developing such applications [17, 18, 19, 5, 20].In early years only one- and two-dimensional data structures were used, due to the limitations of computer resources. The imperative programming paradigm was sufficient for this type of task [18]. With the improvement of computer hardware and the advent of the object-oriented programming paradigm, the shift to more complex data structures was possible. More complexity is added when modeling requires a change of the underlying topological data structure, usually from regular to irregular grids. Additional complexity is introduced by changes in the solver mechanisms or through the use of different types of data, e.g., vectorial or tensorial data [21]. The most drastic changes usually result from a change in the discretization scheme, or the mathematical problem formulation itself that is derived from PDEs [16, 22].

The main motivation for the GTL was the circumstance that a detailed analysis of the tools developed at our institute has shown the following distribution between the amount of source code for data structures and algorithms:

Name	Year	DS	Algorithm	Reference
`MINIMOS`	1980	60%	40%	[16]
`S*AP`	1989	60%	40%	[19]
`MINIMOS-NT`	1996	70%	30%	[5]
`ELSA`	1999	70%	30%	[23]
`WSS`	2000	90%	10%	[20]

Most of these applications use data structures such as `list` and `array` as well as triangles, quadrilaterals, tetrahedra, cuboids, each with their own different access and storage mechanisms, and iteration operations. Although these tools use the C++ STL to some extent, the overall application design is not based on generic libraries. For this reason, the number of source lines is growing quickly due to the complex requirements of two and three-dimensional problems. The currently used applications exceed the limit of maintainability greatly.

This was the start for our own analysis related to data structures and different programming paradigms in TCAD. Our analysis then revealed that, up to now, none of the investigated libraries (BGL, GrAL) can be used directly. For lower-dimensional applications (0D, 1D) the libraries suffer from higher-dimensional information, such as incidence or adjacence. Applications, based on libraries, which use different types of grids (triangles, tetrahedra, cubes) were always outperformed by manually tuned applications. However, for the field of scientific computing, it is essential to abstract from the iteration mechanism, dimensionality, and type of the underlying cell complex.

2.3 Formal Specification

This section introduces the basic notation of topological spaces and cell complexes in our approach. In Figure 3 we present an overview of the terms used.

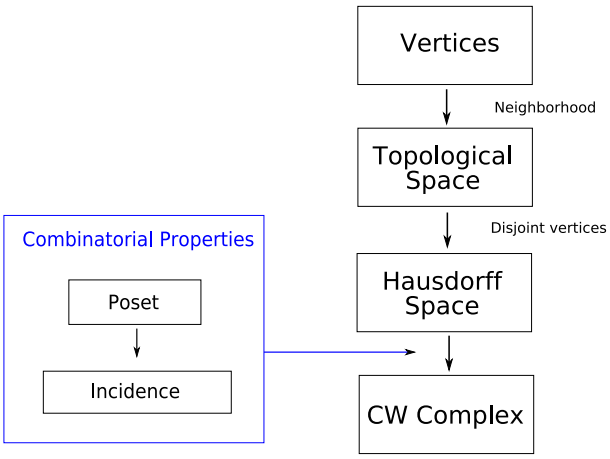

Figure 3: Basic mathematical formalism.

Of particular interest are the combinatorial properties of a CW-complex to characterize different data structures of arbitrary dimensions. Hence, we introduce the formal specification of a CW-complex [24] first. A complete introduction of all terms is available in [9, 24].

Definition: ***CW-Complex*** $\mathcal{C}$***, [24]***
A pair $(\mathcal{T}, \mathcal{E})$, with $\mathcal{T}$ a Hausdorff space and a decomposition $\mathcal{E}$ into cells is called a CW-Complex, iff the following axioms are satisfied:

- mapping function: for each n-cell $c \in \mathcal{E}$ a continuous function $\Phi_e : D^n \rightarrow \mathcal{T}$ exists, which transforms D^n homeomorphically onto a cell c and S^{n-1} in the union of maximal $(n-1)$ dimensional cells. D^n represents an n-dimensional ball and S^{n-1} represents the $n-1$ cell complex.
- finite hull: the closed $\mathrm{hull}(c)$ of each cell $c \in \mathcal{E}$ connects only with a finite number of other cells.
- weak topology: $A \subset \mathcal{T}$ is open, iff each $A \cap \mathrm{hull}(c)$ is open.

An n-cell describes the cell with the highest dimension:

- zero-dimensional (0D) cell complex: vertex
- one-dimensional (1D) cell complex: edge
- two-dimensional (2D) cell complex: triangle

For this work, the most important property of a CW-complex can be explained by the usage of different n-cells and the consistent way of attaching sub-dimensional cells to the n-cells. This fact is covered by

the mapping function. From now, we use an abbreviation to specify the CW-complex with its dimensionality, e.g., a 1-cell complex describes a one-dimensional CW-complex. An illustration of this type of cell complex is given in Figure 4.

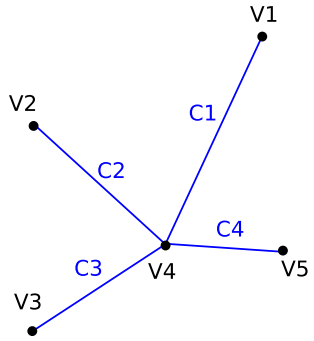

Figure 4: Representation of a 1-cell complex with cells (edges, C) and vertices (V).

In the regime of data structures the requirements of a CW-complex, the finite hull and weak topology, are always satisfied due to the finite structure. The underlying topology of a CW-complex used in computer data structures is always generated from the power set $\mathcal{P}(X)$. For this reason, the topological space cannot be used directly to characterize the different data structural properties. An example is the topological space of a random access container specified by the following code line:

```
std::vector<int> container(3);
```

The topological space $\mathcal{T}$ is described by the power set which models the arbitrary access of this container.

$$\mathcal{T} = \{\emptyset, \{0\}, \{1\}, \{2\}, \{0,1\}, \{0,2\}, \{1,2\}, \{0,1,2\}\}$$

For this reason we introduce the concept of a topological neighborhood [24].

Definition: ***Neighborhood***
A subset $A \subseteq X$ of a topological space $\mathcal{T}$ is a neighborhood of an element $p \in X$, iff it contains an element O of $\mathcal{T}$ that contains p.

$$A \subseteq X \text{ neighborhood of p} \iff \exists \mathcal{O} \in \mathcal{T} : \text{p} \in \mathcal{O}, \mathcal{O} \subseteq \mathcal{A}$$

A *base of neighborhoods* at $p \in X$ is a set of neighborhoods of p such that every neighborhood of p contains one of the base neighborhoods. We introduce the notion of bn which describes the number of elements of the base of neighborhoods. Different data structures can be uniquely characterized by this number. To illustrate this term we present the following list data structure:

```
std::list<int> container(4);
```

$\mathcal{T}$ is also described by the power set but the base of neighborhood can be used to characterize the list. The following sub-set of the topology represents the base of neighborhood of the list:

$$\mathcal{T}_i = \{\{0,1\}, \{1,2\}, \{2,3\}, \{3,4\}\}$$

Next, we introduce the combinatorial properties of a cell complex. These properties are responsible for the internal layout of data structures, as well as for the iteration mechanisms of these data structures.

With the assumption of cell complexes and the base of neighborhoods we introduce the following term:

Definition: ***Adjacence and Incidence***
Given two sets $a, b \in \mathcal{T}$, we define a binary adjacence relation $\mathcal{R}_{\text{adj}}(a,b)$ with the following properties:

$$\mathcal{R}_{\text{adj}}(a,b) : \iff a \cap b \neq \emptyset$$

As a special case of adjacence we define the incidence relation $\mathcal{R}_{\text{in}}(a,b)$:

$$\mathcal{R}_{\text{in}}(a,b) : \iff a \cap b = a \vee b$$

The incidence relation gives the possibility of an iteration of a topological spaces, using only the definition of a base of neighborhoods which separates the combinatorial properties of our underlying topological spaces.

To define higher-dimensional cell complexes, a mechanism is introduced which handles the internal structures of cells. The topological space of, e.g., a triangular grid is described by the vertex on cell information. The number of elements of a sub-set does not give any information about the internal structure of this element. The sub-set $\mathcal{T}_j = \{1,2,3,4\}$ can describe a tetrahedron in three dimensions or a quadrilateral in two dimensions. In order to be able to distinguish these different element, we introduce the concept of a poset:

Definition: ***Poset, [25]***
A poset ($\mathcal{S}$,<) is a finite set $\mathcal{S}$, together with a partial order relation.

In the case of a cell complex, the partial order relation is described by incidence. A Hasse diagram can be used to visualize the poset of a cell. Any two elements are connected by a line, if they are comparable.

With the Hausdorff property of the CW-complex we can uniquely characterize cells or faces by their set of vertices. We define $\{a,b,c\}$ as the element which exactly contains the vertices a, b, c.

Another important property is the locality of the cell complex. Two different properties can be distinguished, which represent the arbitrary and the iterative access of data structures.

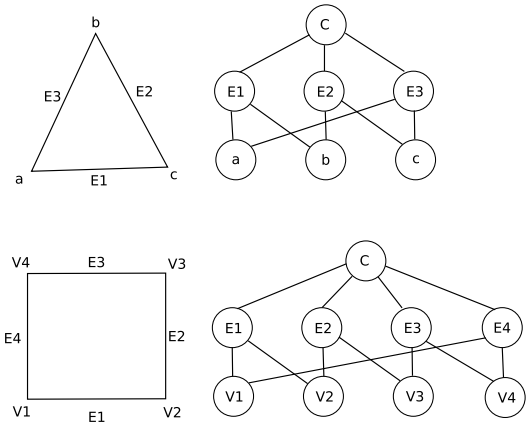

Figure 5: A Hasse-diagram for a triangle cell (top) and a quadrilateral cell (bottom).

Definition: ***Global Cell Complex***
A cell complex $\mathcal{C}$ which is homeomorphic to the following combinatorial structure of cells [9], where i_d represents the dimensional ticks:

$$\{[i_1, i_1 + 1] \times .. \times [i_d, i_d + 1] \mid 0 \leq i_j \leq m_j\}$$

is called a cell complex with *global* properties. Here the topological incidence relation is apparent from the fact that global information is explicitly available. This property is important because of the fact, that a global cell complex describes the *random access* container types.

Definition: ***Local Cell Complex***
Conversely, a cell complex which cannot be described globally is called a local cell complex.

In scientific computing, neighborhood information of a local cell complex has to be stored explicitly. Due to the non-trivial construction of instances of cell complex types, we refer to literature [26]. Related to data structures, a local cell complex models different types of lists, trees, or maps.

2.4 Generic Topology Library

In this section we introduce the basic idea of the underlying cell complex for data structures. The classification of each data structure is using the dimension of the cells. Figure 6 shows a 0-cell complex. In this special case, cells and vertices are identical. No neighborhood information is given, only the cells are depicted. This topological structure covers most of the STL data structures. The differences between each of the data structures such as `std::vector` and `std::list` can be found in the *base of neighborhoods* and the *incidence relation* or, in other words, in their combinatorial properties.

The internal mechanism and utilization of the internal structure of the data structure is not possible due to the 0-cell complex, which means that no higher incidence or adjacence (see Section 2.3) is available directly. No data can be stored on edges or cells easily.

Generic algorithms cannot always use the internal structure of, e.g., `std::map` or `boost::graph` without modification. Copying a map or graph could be much more efficient, if the algorithms were aware of different internal topologies of data structure, such as the tree structure of a map.

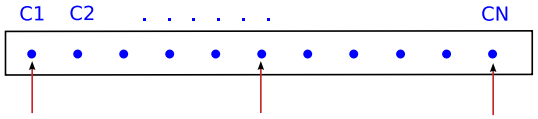

Figure 6: Iteration over cells within a 0-cell complex.

As already mentioned, applications designed in the field of scientific computing need higher-dimensional data structures as well as higher-dimensional iteration operations. Consider, for example, a 1-cell complex (Figure 7) and a 2-cell complex (Figure 8).

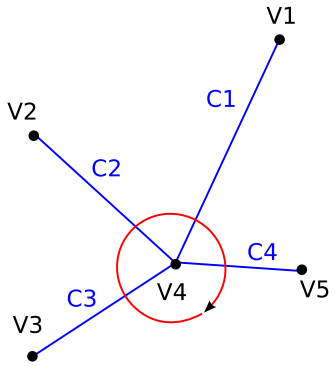

Figure 7: Iteration over edges for a 1-cell complex.

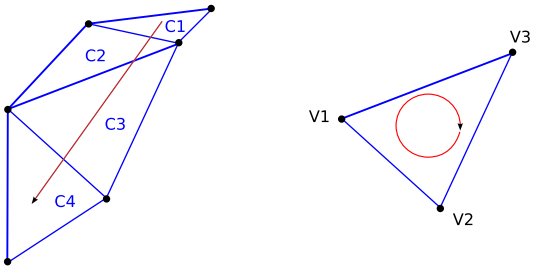

Figure 8: Iteration over cells and incident vertices of a cell for a 2-cell complex.

In the case of the 1-cell complex, the basic concept of *incidence* is mostly covered. There are only edges and vertices, and most of the operations on these two elements can be implemented with basic methods. For higher-dimensional cell complexes, e.g., a 2-cell complex, the

incidence relation becomes more complex. There are various permutations of incidence relations which all lead to a different iteration. All vertices connected to a triangle, or all edges which are part of the triangle can be traversed. Also adjacent iteration can be derived easily.

2.4.1 Topological Properties of Data Structures

We can now show, based on the formal definitions in Section 2.3, that we can derive a consistent categorization of different data structures and therewith a homogeneous interface which does not restrict the dimensionality or iteration mechanism of the data structures. In the following table we characterize common data structures with their combinatorial properties. The used terms are:

- dim: dimension of the cell complex
- locality: refers to the local or global combinatorial properties of the underlying space
- bn: represents the number of the elements of the base of neighborhoods of the cell

SLL stands for single-linked-list whereas DLL means double-linked-list. A global defined cell complex does not require a base of neighborhood due to the fact, that the neighborhood is implicitly available.

data structure	dim	locality	bn
array/vector	0	global	
SLL/stream	0	local	2
DLL/binary tree	0	local	3
arbitrary tree	0	local	4
graph	1	global	
regular grid	2	global	
irregular rid	2	local	4
regular grid	3	global	
irregular grid	3	local	5

The next code snippet presents our topological data structure definition. The first number stands for the actual dimension, the tags *global* and *local* stand for the combinatorial property, and the final number specifies the number of elements of the base of neighborhoods.

```
topology<0,global > topo;// array
topology<0,local,2> topo;// SLL/stream
topology<2,global > topo;// regular grid
topology<2,local,4> topo;// irregular grid
```

For a 0-cell complex the STL iterator traits can be used to classify the data structure easily:

```
topology<0,random_acess>  topo;// global
topology<0,forward>       topo;// local,2
topology<0,bidirectional> topo;// local,3
```

Based on this formulation, an automatic mechanism is possible to derive optimal data structures based on the requirements of algorithms.

To show the implementation with the GTL and equivalence of the data structure compared to the STL vector a simple code snippet is presented:

Equivalence of data structures

```
typedef topology<0, random_access> topo_t;
typedef long data_t;

typedef cell_t<topo_t, data_t> container_t;
container_t container;

// is equivalent to

std::vector<data_t> container;
```

Here, the separation of the topological structure specification can be clearly observed.

2.4.2 Finite Cell Complexes

This section deals with the analysis of the data structures from the STL and BGL and generalize these expressions to arbitrary-dimensional data structures. We show that all different data structures model a common interface and each dimension can use specializations to obtain the best performance.

The 0-Cell Complex A typical representative of a 0-cell complex is the topological structure of a simple array. The C++ STL containers such as `vector` and `list` are representatives and are schematically depicted in Figure 9. The points represent the cells on which data values are stored.

Figure 9: Representation of a 0-cell complex with a topological structure equivalent to a standard container.

Iteration and data access is used simultaneously in the basic iterator concept of the STL. The next code snippet presents these facts, where the forward iteration `++it` is used to traverse the cells. The `*it` is used to access the value attached to the cell at position `it`.

C++ STL approach

```
std::vector<int> container;
std::vector<int>::iterator it;

it = container.begin();
++it;              //topological traversal
int value = *it;   //data access
```

On the one hand side, the iterator concept is one of the key elements of the C++ STL. It separates the actual data structure access from algorithms. Thereby the implementation complexity is significantly reduced. On the other hand side, it combines iteration and data access. The improvements of separating the iteration and data access are outlined with a cursor and property map concept [11]. A possible application of this approach is demonstrated in the next code snippet:

Separated iteration and data access

```
vector<bool> container;
vector<bool>::iterator it;
property_map           pm(container);

it = container.begin();
++it;                  //iteration
bool value = pm(*it);  //data access
```

The `std::vector<bool>::iterator` can be modeled by a random access iterator [11], whereas the data access returns a temporary object which can be used efficiently [27] with modern compilers. Additionally, this mechanism offers the possibility of storing more than one value corresponding to the iterator position. This feature is especially useful in the area of scientific computing, where different data sets have to be managed, e.g., multiple scalar or vector values on a vertex, face, or cell.

Based on the formal classification of Section 2.3 we analyze the combination of iteration and data access in more detail. The following list overviews the basic iterator traits [28]:

- input/output
- forward
- bidirectional
- random access

As we have seen, there is a unique and distinguishable definition possible for all of these data structural properties. On the one hand side, the backward and forward compatibility of the new iterator categories are a major problem [29]. On the other hand side, problems are encountered, if we integrate the iterator categories into our topological specification. In the following the replacement for the input and output traits is listed:

- incrementable
- single pass
- forward

The combinatorial property of the underlying space of these three categories is the same: a 0-cell complex with a local topological structure, defined by the following code snippet:

```
topology<0, local, 2>  tp;
```

The old iterator properties have only used two different categories which specify the data behavior, namely the input and output property.

The difference between these three categories can be described by:

- incrementable: this is a topological property only
- single pass: this is a data property only
- forward: this combines the incrementable and single pass properties

Only the incrementable property can be described by a topological property, whereas the other two categories are data dependent.

The 1-Cell Complex This type of cell complex is usually called a graph. Figure 4 presents a typical example. A cell of this type of cell complex is called an *edge*. Incidence and adjacence information is available between edges and vertices.

We give examples on simple algorithms based on graphs using the BGL. The BGL implements comprehensive and high performance graph treatment capabilities including the associated adjacence and incidence relation. Iteration and data access are separated by the already mentioned cursor and property map concept [10]. The next code snippet presents an iteration using mechanisms of the BGL. In this algorithm all edges are traversed.

BGL iteration

```
typedef adjacency_list<vecS, vecS> Graph;
Graph gr(number_of_points);

// edge initialization

edge_iterator eit, eit_end;

for (tie(eit, eit_end) = edges(gr);
     eit != eit_end; ++eit)
{
  test_source1 += source(*eit, gr);
  test_source2 += target(*eit, gr);
}
```

With the GTL the same functionality can be accomplished as demonstrated in the following code snippet. The `global` keyword is used to highlight the global structure of the graph, which means, that the internal data layout is prepared for a dense graph storage.

GTL iteration

```
typedef topology<1, global> topo_t;
topo_t topo(number_of_vertices);

// cell initialization

cell_on_vertex_it covit, covit_end;

for (tie(covit, covit_end) = cells(topo);
     covit != covit_end; ++covit)
{
  test_source1 += source(*covit, topo);
  test_source2 += target(*covit, topo);
}
```

The ND Cell Complex We extend the 0-cell and 1-cell complex types to arbitrary-dimensional cell complexes. In this work we restrict the topological spaces to the most important to scientific computing: the local (Figure 10) and the global cell complex (Figure 11). Based on our cell complex types the following cell types are available:

- 0-cell: vertex
- 1-cell: edge
- 2-cell: triangles, quadrilaterals
- 3-cell: tetrahedra, cubes

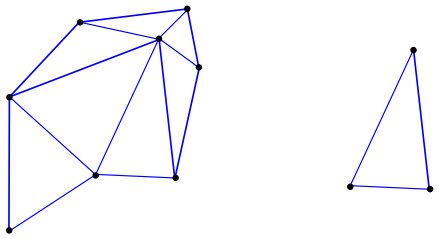

Figure 10: Local cell complex (left) and a cell representation (right). Vertices are marked with black circles.

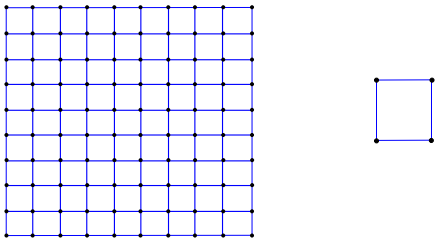

Figure 11: Global cell complex (left) and a cell representation (right).

The following code snippet presents the implementation of an arbitrary topology with the structure of a local 2-cell complex. The stored data is based on scalar values using a `double` for representation.

Iteration with our approach

```
typedef topology<2, local, 4>          topo_t;
topology_traits<topo_t>::iterator      it;

typedef data<scalar, double>           data_t;
data_t                                 data;
data_traits<data_t>::value             value;

it = topo.vertex_begin();

++it;                          // iteration
value = data(*it);             // access
```

The next example presents an iteration mechanism starting with an arbitrary cell iterator evaluated on a cell complex, which is an instance of a topological cell complex. Then a vertex on cell iterator is initialized with a cell of the complex. The iteration is started with the `for` loop. During this loop an edge on vertex iterator is created and initialized with the evaluated vertex. This edge iterator starts the next iteration. The corresponding graphical representation is given in Figure 12. The necessary *valid()* mechanism models a *circulator* concept [30]. The objects marked depict the currently evaluated objects. In the first iteration state the vertex `v1` is used and the iteration is performed over the incident edge, then the iteration continues with the remaining vertices.

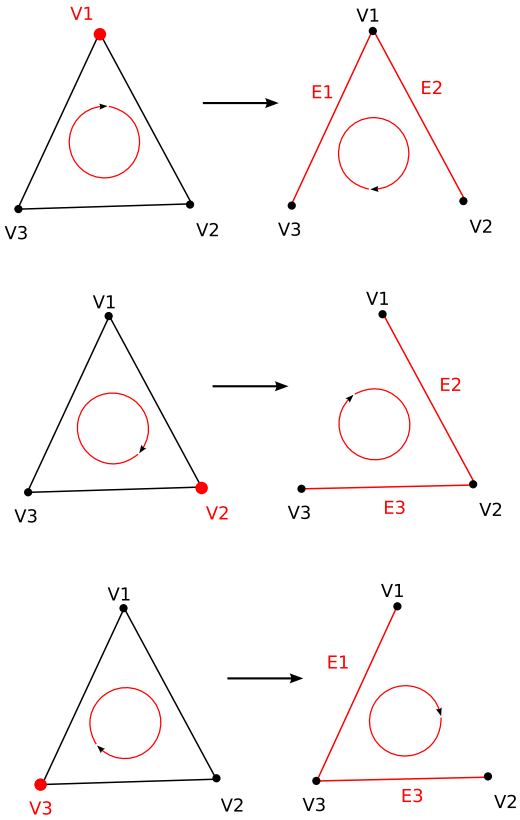

Figure 12: Incidence relation and iteration mechanism.

A more complex iteration

```
cell_iterator ce_it = topo.cell_begin();

vertex_on_cell_iterator vocit(*ce_it);
for(; vocit.valid(); ++vocit )
{
  edge_on_vertex_iterator eovit(*vocit);

  for (; eovit.valid(); ++eovit )
  {
     //operations on edges
  }
}
```

As can be seen, the iteration mechanism can be used independently of the used dimension or type of cell complex. The iteration is initialized with a cell iterator only. Three different objects have to be assured by the cell complex: vertices, edges, and cells. All cell complex types which support these three objects can be used for this iteration.

2.5 Data Access

We use the *property map* concept by a functional access mechanism called *data accessor*. The data accessor implementation also takes care of accessing data sets with different data locality, e.g., data on vertices, edges, facets, or cells. This locality is specified by the given key `key_d`. During initialization the data accessor `da` is bound to a specific cell complex with that key. The `operator()` is evaluated with a vertex of the cell complex as argument. The next code snippet presents this assignment briefly.

Data assignment

```
string key_d  = "user_data";
data_t da = scalar_data(topo, key_d);

da(vertex) = 1.0;
```

In the following code snippet, a simple example of the generic use of this accessor at run-time is given, where a scalar value is assigned to each vertex in a domain. The data accessor creates an assignment which is passed to the `std::for_each` algorithm.

Data assignment

```
da_t da = scalar_data(topo, key_d);

for_each
(
 topo.vertex_begin(),
 topo.vertex_end(),
 da = 1.0
);
```

Another example is given, where the data accessor is combined with the topological structure to completely specify a container. The data accessor can be used independently.

Equivalence of data structures

```
typedef topology<0, random_access> topo_t;
typedef long data_t;
container_t<topo_t, data_t> container;

// is equivalent to

std::vector<data_t>       container;
```

2.6 GTL Architecture

The GTL is based on a layered concept, which means that the iteration mechanism and data access mechanisms are orthogonal (Figure 13). The lowest layer represents the concepts for cell, vertex, and the poset information. The other part of the lowest layer implements the data storage. It can be observed that data can be handled independently of the topological information and iteration. The second layer provides the incidence relation and the data accessor mechanisms.

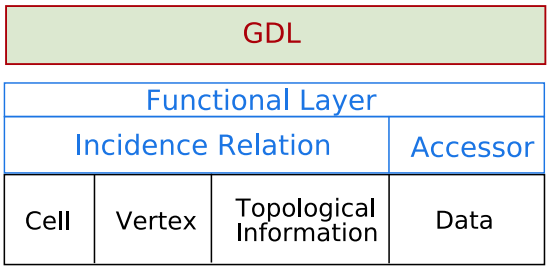

Figure 13: Conceptual view of the GTL.

The highest level in the GTL is based on meta and functional programming for a convenient usage of the different iteration mechanisms. To illustrate these mechanisms different examples are presented. The first snippet shows a simple functional iteration:

Functional iteration

```
typedef topology<2, random_access> topo_t;
typedef long data_t;
container_t<topo_t, data_t> container;

gtl::iterate<vertex_on_cell>
[
  std::cout << _1 << std::endl
](container);
```

With the GDL, different algorithms can be used as well as presented in the next example. Here different topological containers can be traversed and the data is accumulated and printed.

GTL iteration with GDL mechanisms

```
typedef topology<2, random_access> topo_t;
typedef long data_t;

container_t<topo_t, data_t> container;
container_t::data_accessor da;

gtl::iterate<cell>
[
   std::cout <<
   gdl::sum<vertex_on_cell>(0.0)[da(_1)]
  << std::endl
](container);
```

2.7 Outlook

The field of scientific computing requires an efficient notation of equation systems, has to construct equations, and has to abstract from the iteration mechanisms of different underlying objects. Various algorithms in the field of scientific computing only depend on the combinatorial properties of the underlying space. Using only combinatorial information results in more stable algorithms.

By providing a concise interface to different kinds of data structures, a new type of equation specification is made possible. In this way algorithms and equations can be specified independently from dimension or topological cell complex types.

To show the requirement for the equation specification we use a simple equation system resulting from a self-adjoint PDE type. Figure 14 presents a local patch of a 1-cell complex on which the equation is evaluated.

The data `Aij` represents the area of the dual graph (Voronoi graph). Using a finite volume discretization scheme [16] a generic Poisson equation $\mathrm{div}(\varepsilon\ \mathrm{grad}(\Psi)) = \varrho$ can be formulated in two spatial

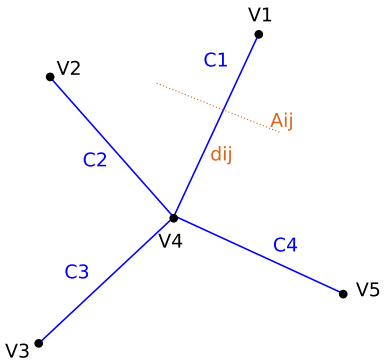

Figure 14: Cell complex with corresponding data.

dimensions as:

$$\sum_j D_{ij}\ A_{ij} = \varrho \tag{6}$$

$$D_{ij} = \frac{\Psi_j - \Psi_i}{d_{ij}}\ \frac{\varepsilon_i + \varepsilon_j}{2} \tag{7}$$

D_{ij} stands for the projection of the dielectric flux onto the cell/edge c_i that connects the vertices v_i and v_j. The direct transformation of each equation element can be observed clearly when considering the following source code:

Generic Poisson equation

```
value =
(
  gdl::sum<vertex_edge>
  [
     gdl::diff<edge_vertex>
     [
       Psi(_1)
     ] * A(_1)/d(_1) *
     gdl::sum<edge_vertex>[epsilon(_1)] / 2
   ] - rho(_1)
)(vertex);
```

The term `Psi` represents the distributed data set, `A` the Voronoi area, `d` the distance of two points, `rho` the right hand side, and `epsilon` some material property. It is important to stress that all data sets have to be evaluated in their right data locality, that is `Psi`, `epsilon`, and `rho` on vertices and `A`, `d` on the incident edges. The example uses the unnamed function object `_1` only. The data accessor implementation handles the correct access mechanism. The GDL implements mechanisms to derive the correct data locality of each unnamed object. An in-depth discussion is given in [15] The complex resulting from this mapping is completed by specifying the current vertex object `vertex` at run-time.

2.8 Conclusion

We have shown that the specific properties of different data structures can be specified by means of topological and combinatorial properties. An automatic derivation of optimal data structures based on the requirements of algorithms is possible.

Based on the topological properties the iterator traits can be derived automatically from combinatorial properties of the corresponding data structure. A concise iteration mechanism for different dimension is presented which includes the STL containers as well as higher-dimensional cell complex types. Different issues of the currently used iterator mechanism can be easily explained.

The full power of functional programming is revealed, when it performs different types of topological traversal with our approach.

3 Heatring - Smart Investigation of Temperature Impact on Integrated Circuit Devices

Abstract

To investigate the electrical on-chip-transistor behavior at different temperatures usually the transistor area on the wafer is heated by external heat sources to operate at a specific temperature. To avoid using external heat sources a heatring structure was developed which directly controls the temperature of the investigated transistor area on the wafer, guaranteeing very fast warming up and cooling off duration times. Testing the heatring functionality was performed by electro-thermal simulations, the results of which were verified by measurements.

3.1 Motivation

For testing the reliability and temperature dependence of semiconductor devices it is necessary to operate these devices at different temperature. One possibility is to use an external heat source such as a thermo-chuck. A smart option is to use a heatring structure. This heatring structure is placed in the wafer around the test region. Electrical power loss in the heatring structure directly heats the small well defined test area. The very big advantage is that the heating process of this small device area has a duration of only a few milliseconds. The thermal stress stays locally limited. The thermal energy produced by the electric current is well tunable. For electrically analyzing a device within the heatring, the temperature profile inside the heatring area should stay constant. Therefore, in this work we investigate the temperature distribution in the heatring area by a given applied voltage and operating time by electro-thermal simulations. For the evaluation of these simulations the simulated temperature results are compared to measured values.

This heatring structure is patent pending.

3.2 Investigated Structure

The structure is implemented in a WLR-monitor (Wafer Level Reliability) or SLM (Scribe Line Monitor). A SLM module is used to validate the manufacturing process. It can consist of simple structures like resistors, capacitors, NMOS and PMOS transistors or even just simple rectangles to validate the lithography process step. A SLM is placed in the scribe line (the scribe line is the "virtual" line where the wafer will be sawed). The SLM contains two heatrings where two different PMOS transistors with different gate width are placed inside (Fig. <15>). The simulation of both heatrings is not necessary, since for both heatrings at the designated temperature the results would be the same. Here, only the temperature distribution over the heatring structure is simulated. The analysis of the electrical properties of the transistors is not the scope of this work and can be performed by a device simulator (Minimos-NT [5] or Dessis [31], for instance).

Fig. <16> shows the experimental heatring structure on the SLM. NDIFF is a high doped n-type area which represents the real heatring. SN is a middle doped n-type area. The test device is placed in the n-well DN, shielded from the former n-doped structures by the p-doped protection ring RP. These structures are embedded in the silicon wafer. Only the metallic supply pad M1 lies above the wafer. However, for the electro-thermal simulation it is sufficient to consider only the conducting heatring and its insulating environment with the corresponding thermal properties.

3.3 Thermal Simulation and Measured Results of the Heatring SLM

The temperature distribution which occurs directly on the wafer is measured by diodes (Fig. <17>) placed within the heatring. Fig. <18> illustrates the simulated temperature distribution within the heatring structure at $70V$. In this figure the top layer which consists of SiO_2 is removed for visualization purposes. However, in the simulation this oxide layer must not be neglected to take into account the materials around the heatring. The top of the oxide layer is exposed to room temperature (300K), corresponding to thermal Dirichlet boundaries. All remaining outer faces are adiabatic (zero Neumann boundary conditions). For the electrical problem at the outer faces of the simulation area zero Neumann boundary conditions are applied. The electrical Dirichlet boundary conditions are represented by the control voltage applied at the metal pads which are connected by vias to the middle of the left side of the heatring and to the middle of the right side, respectively. The simulation is performed by our in house interconnect simulation software *Smart Analysis Programs* [32]. It is based on the Finite Element Method [33] on tetrahedral grid elements [34]. The rise of the temperature from the center to the heat source is about the same as with the measured result shown in Fig. <17>. Fig. <19> shows the results of the simulated and measured temperature in the center of the heatring structure. Only at higher heatring temperatures the curves are slightly different. At 90V the simulated curve gets more than 160°C. Since the simulated structure is very small compared to a wafer, the

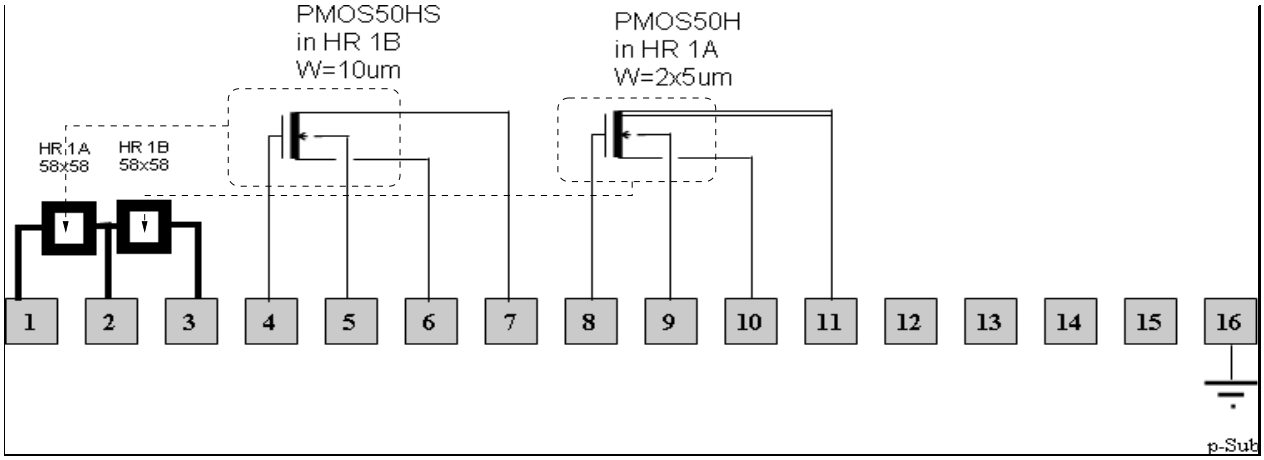

Figure 15: Schematic of the heatring SLM structure

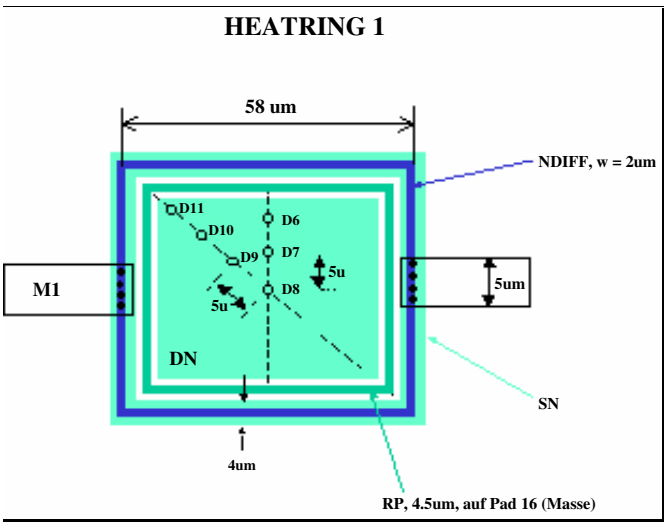

Figure 16: Heatring structure

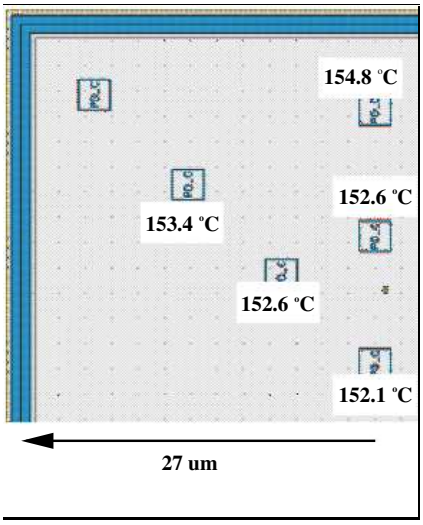

Figure 17: Measured heatring temperature distribution at $90V$

temperature distribution is "cut" on the side areas (homogeneous Neumann boundary conditions). So the heat-flow normal to the surface is stopped and thus the temperature of the whole model region is higher than expected. On the real wafer the heat-flow is not limited close to the heat source. Thus there is enough space for a wider heat spread and the measured curve is almost linear at higher heatring voltages. Enlarging the thermal simulation area overcomes this constraint, however, in this case longer simulation durations have to be accounted for.

3.4 Theoretical Background

3.4.1 Electro-Thermal Simulation

For a coupled electro-thermal simulation the heat conduction system

$$c_p \rho \frac{\partial T}{\partial t} - \vec{\nabla} \cdot (\gamma_T \vec{\nabla} T) = p \tag{8}$$

has to be considered [35] [36], where the solution of (8) gives the temperature T. The material properties are defined by the thermal conductivity γ_T, by the specific heat c_p, and by the mass density ρ. The source density function p corresponds to the electrical power loss density and is calculated by

$$p = \gamma_E (\vec{\nabla} \varphi)^2. \tag{9}$$

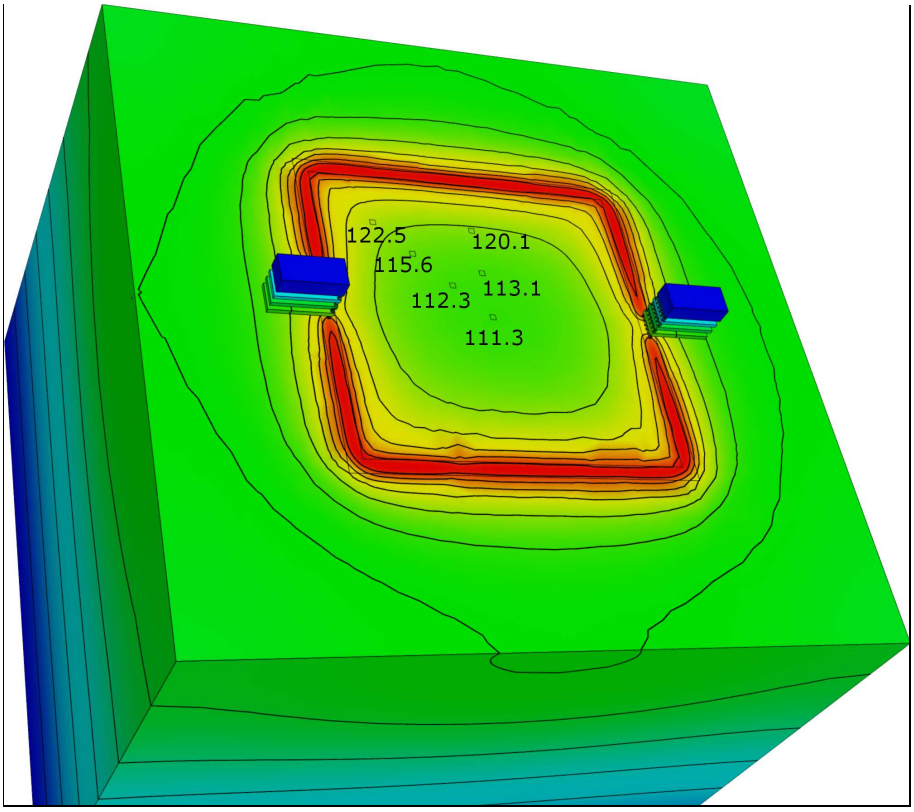

Figure 18: Simulated heatring temperature distribution at $U_{\mathrm{Heatring}} = 70V$

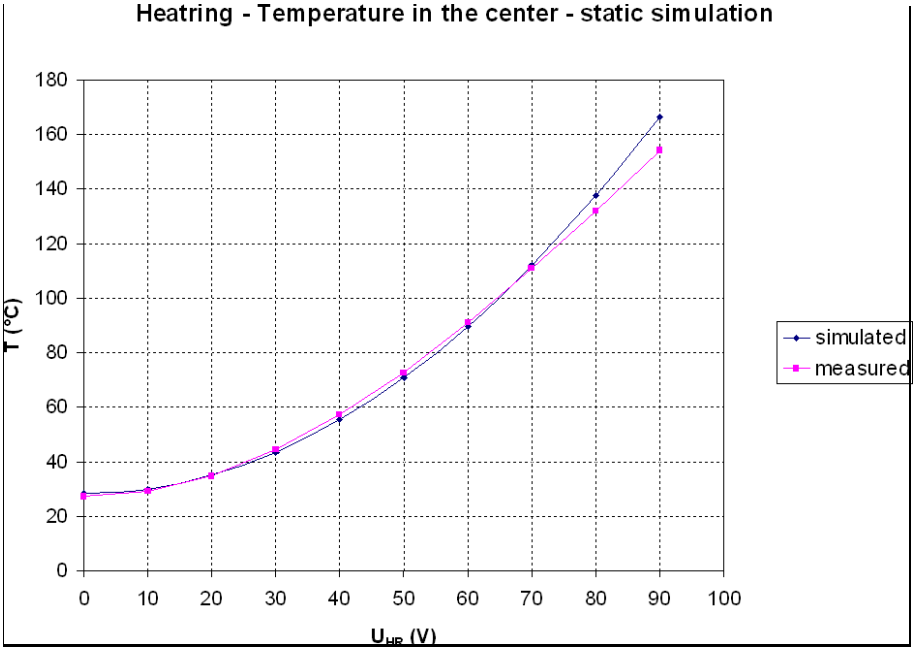

Figure 19: Temperature in the heatring center

Finally, the power loss density p in (9) is derived from the the electric potential φ, which is calculated by solving the Euler equation

$$\vec{\nabla} \cdot (\gamma_E \vec{\nabla} \varphi) = 0, \tag{10}$$

where γ_E denotes the electrical conductivity. The partial differential equations (8), (9) and (10) couple the electrical and the thermal system.

The electrical conductivity and the thermal conduc-

tivity of most materials depend on the temperature. Usually the following model is used to describe this dependence:

$$\gamma(T) = \gamma_0 \frac{1}{1 + \alpha(T - T_0) + \beta(T - T_0)^2}.$$

γ_0 is the (electric/thermal) conductivity at reference temperature T_0 (300K). α and β denote a linear and a quadratic temperature coefficient, respectively.

3.4.2 Boundary Conditions

The boundary of the simulation area G is divided into two parts for the thermal system (G_{T1} and G_{T2}) and also into two parts for the electrical system (G_{E1} and G_{E2}). The Dirichlet boundary conditions for the thermal part of the system model the heat sinks.

$$T = T_c \quad \text{on } G_{T1}$$

An adiabatic (ideal thermally insulating) boundary is described by homogeneous Neumann boundary conditions

$$\vec{n} \cdot \vec{\nabla} T = 0 \quad \text{on } G_{T2}.$$

The applied electrical contact potentials φ represent Dirichlet boundary conditions for the electric part

$$\varphi = \varphi_c \quad \text{on } G_{E1}.$$

Constant current sources are implemented by Neumann boundary conditions

$$\vec{n} \cdot \vec{\nabla} \varphi = f_c \quad \text{on } G_{E2} \quad \text{with } J_n = \gamma \, \vec{n} \cdot \vec{\nabla} \varphi.$$

3.4.3 Initial Conditions

For transient thermal problems the condition T_0 for the temperature at initial time $t = 0$ has to be defined

$$\forall \vec{r} \in \mathcal{V},\ t = 0 :\ T(\vec{r}, 0) = T_0.$$

3.5 Conclusion

In this work a heating structure placed in a wafer around a test region is proposed. This structure guarantees constant temperature distribution in the investigated area and a very fast heating process.

4 Monte Carlo Study of Electron Transport in Strained Silicon Inversion Layers

The effect of degeneracy both on the phonon-limited mobility and the effective mobility including surface-roughness scattering in unstrained and biaxially tensile strained Si inversion layers is analyzed. We introduce a new method for the inclusion of the Pauli principle in a Monte Carlo algorithm. We show that incidentally degeneracy has a minor effect on the bulk effective mobility, despite non-degenerate statistics yields unphysical subband populations and an underestimation of the mean electron energy. The effective mobility of strained inversion layers slightly increases at high inversion layer concentrations when taking into account degenerate statistics.

4.1 Introduction

During the last years the introduction of strain in the channel of Si MOSFETs has become a widely used technique to improve transistor drive currents [37] [38] [39]. Strain can be induced by epitaxially growing thin Si layers on relaxed $Si_{1-y}Ge_y$ substrates, or alternatively, by processing additional cap layers over the transistors. The latter method is especially suitable for mass production because it requires only a slight modification of the process flow [40].

Surprisingly, from a theoretical viewpoint the mobility enhancement caused by strain is still an issue of discussion. The reason for this is manifold: It was claimed that using the well established models for scattering in the two dimensional electron gas (2DEG) the mobility gain of strained Si (SSi) at low effective fields should be compensated by more pronounced surface roughness scattering at large effective fields [41]. Due to this fact, only with the assumption of much smoother strained Si-SiO_2 interfaces one should be able to get qualitative agreement with experimental data. Even though it seemed to be unphysical to change the smoothness of the strained Si-SiO_2 interface, the Monte Carlo (MC) community has adopted this assumption [42] or simply not responded to the troubling fact.

When trying to clarify this dissatisfactory status two main difficulties arise: First, there exists a variety of surface roughness scattering models, and it is not clear which approximations to the general expression given by Ando [43] are allowed. Second, there is a discordance whether and how degeneracy effects should be included in transport calculations of inversion layers.

In this paper, the ways to include the Pauli principle in a MC algorithm are revised and critically compared to each other. The usual method, where the Pauli blocking factor $1 - f(\mathbf{k})$ is approximated using the equilibrium distribution function $f_{\mathrm{FD}}(\mathbf{k})$, can be shown to lead to unphysical subband populations, kinetic energies, and mobilities. The reason being that at high degeneracy the error $\varepsilon(\mathbf{k}) = f(\mathbf{k}) - f_{\mathrm{FD}}(\mathbf{k})$ is dominant. A new MC algorithm accounting for the Pauli exclusion principle is proposed which is less sensitive to the error $\varepsilon(\mathbf{k})$.

The paper is organized as follows: Section II describes the new approach to implement the Pauli exclusion principle in the MC method. It is shown that in the low field limit the proposed algorithm yields the same mobility as the Kubo Greenwood formula, while other algorithms do not. We use the new method to extract velocity profiles and illustrate the large effect of degeneracy on the electron system. Finally, the simulated effective mobility curves for unstrained and biaxially tensile strained Si on relaxed $Si_{1-y}Ge_y$ substrates are presented in Section III.

4.2 Inclusion of the Pauli Principle in Monte Carlo Simulations

In transport calculations of the 2DEG forming in the channel of MOSFETs the inclusion of the Pauli principle is expected to be important since the lowest subband may lie well below the Fermi level in the regime of moderate and high effective fields (high inversion layer concentrations). This leads to modified subband populations and an elevated mean kinetic energy of electrons as compared to the nondegenerate case. A change in the mobility is therefore to be expected.

The proposed algorithm is based on the following reformulation of the degenerate scattering operator.

$$\begin{aligned} Q[f]_{\mathbf{k}} &= \int f_{\mathbf{k}'}(1-f_{\mathbf{k}})S_{\mathbf{k}',\mathbf{k}}\,d\mathbf{k}' - \int f_{\mathbf{k}}(1-f_{\mathbf{k}'})S_{\mathbf{k},\mathbf{k}'}\,d\mathbf{k}' \\ &= \int f_{\mathbf{k}'}S_{\mathbf{k}',\mathbf{k}} - f_{\mathbf{k}}S_{\mathbf{k},\mathbf{k}'} + \underbrace{f_{\mathbf{k}}f_{\mathbf{k}'}(S_{\mathbf{k},\mathbf{k}'} - S_{\mathbf{k}',\mathbf{k}})}_{\text{additional term}}\,d\mathbf{k}'. \end{aligned}$$

The last term represents a nonlinear correction to the non-degenerate scattering operator. To linearize the scattering operator it is common to keep one factor of the product $f_{\mathbf{k}}f_{\mathbf{k}'}$ constant and to treat the other as the unknown.

Near thermodynamic equilibrium, f can be approximated by the Fermi-Dirac distribution function f_{FD}. The key point of the new method is that a symmetric approximation with respect to $\mathbf{k}$ and $\mathbf{k}'$ is employed.

$$f(\mathbf{k})f(\mathbf{k}') \approx \frac{1}{2}\big(f(\mathbf{k})f_{FD}(\mathbf{k}') + f_{FD}(\mathbf{k})f(\mathbf{k}')\big) \quad (11)$$

Using this approximation the scattering operator can be expressed in terms of a modified transition rate $\widehat{S}_{\mathbf{k},\mathbf{k}'}$ and

scattering rate $\widehat{\lambda}_{\mathbf{k}}$ as

$$Q[f]_{\mathbf{k}} = \int f(\mathbf{k}')\widehat{S}_{\mathbf{k}',\mathbf{k}}\, d\mathbf{k}' - f(\mathbf{k})\widehat{\lambda}_{\mathbf{k}}.$$

with

$$\widehat{S}_{\mathbf{k},\mathbf{k}'} = S_{\mathbf{k},\mathbf{k}'}\left(1-\frac{1}{2}f_{FD}(\mathbf{k}')\right) + S_{\mathbf{k}',\mathbf{k}}\frac{1}{2}f_{FD}(\mathbf{k}') \quad (12)$$

$$\widehat{\lambda}_{\mathbf{k}} = \int \widehat{S}_{\mathbf{k},\mathbf{k}'}\, d\mathbf{k}' \quad (13)$$

A simple error analysis shows the advantage of this formulation. Consider a highly degenerate state $\mathbf{k}$, characterized by $f(\mathbf{k}) \approx 1$. A direct approximation of the blocking factor $(1 - f(\mathbf{k}))$ can give completely wrong results, because the approximation of the blocking factor is determined by the error, $1 - (f_{FD} + \varepsilon) \approx \varepsilon$. In the formulation (12), however, because of $\varepsilon \ll 1$ the effect of the error will be negligible, $1 - (f_{FD} + \varepsilon)/2 \approx 1/2$.

The modified transition rate (12) is given by a linear combination of the forward rate $S_{\mathbf{k},\mathbf{k}'}$ and backward rate $S_{\mathbf{k}',\mathbf{k}}$. The latter can be expressed in terms of the forward rate by means of the principle of detailed balance. The modified scattering rates for phonon emission and absorption become,

$$\widehat{\lambda}_{\mathrm{em}} = \lambda_{\mathrm{em}} \cdot \left(1 - \frac{1}{2}\frac{f_{\mathrm{FD}}(\epsilon_f)}{N_0+1}\right)$$

$$\widehat{\lambda}_{\mathrm{ab}} = \lambda_{\mathrm{ab}} \cdot \left(1 + \frac{1}{2}\frac{f_{FD}(\epsilon_f)}{N_0}\right),$$

where ϵ_f denotes the final energy and N_0 the equilibrium phonon distribution function,

$$N_0 = \frac{1}{\exp\left(\frac{\hbar\omega_0}{\mathrm{k_B}T}\right) - 1}.$$

For elastic scattering mechanisms the modified scattering rates do not change from the classical ones, $\widehat{\lambda}_{\mathbf{k}} = \lambda_{\mathbf{k}}$. In the simulation of the 2DEG to a good approximation one can assume scattering with surface roughness, impurities, and acoustical phonons to be elastic.

To implemenent the Pauli principle in a conventional MC program for non-degenerate statistics the only modifications necessary are the replacement of the classical scattering rates by the modified ones.

4.2.1 Comparison of Algorithms

The new algorithm has been compared to two other methods found in the literature [44, 45]. The first algorithm to include the Pauli principle in the MC technique [44] is based on a self-consistent iterative algorithm that uses a rejection technique to account for the occupation probability of the final state at each scattering event. Since this auxiliary self-scattering mechanism is proportional to the occupation of the final states, the algorithm prevents a large number of classically allowed transitions.

A different approach to include degeneracy in MC simulations was given in [45]. Inelastic scattering rates are multiplied with a factor of $(1 - f_{\mathrm{FD}}(\epsilon_f))/(1 - f_{\mathrm{FD}}(\epsilon_i))$, where ϵ_i (ϵ_f) denotes the initial (final) electron energy. This additional factor stems from the use of Fermi-Dirac statistics within the relaxation time approximation [46].

In the limit of vanishing field the mobility can also be calculated using the relaxation time approximation. Thus we compare our MC algorithm and a modified method from [44] and [45] to the mobility calculated from the Kubo-Greenwood expression [47]. From Figure 20 it can be seen that the new method yields the closest agreement, whereas a non-selfconsistent implementation of the algorithm proposed in [44], where $f(\mathbf{k})$ has been approximated by the equilibrium distribution function $f_{\mathrm{FD}}(\mathbf{k})$, and the algorithm proposed by [45] overestimate the effective mobility.

4.3 Results

We extract the mean electron velocity as a function of total electron energy in the small-field limit, by recording the velocity component along the driving field as a function of the total electron energy. For this purpose the particles energy domain was divided into a set of intervals $\Delta\mathcal{E}$. The mean velocity of an electron in a particular interval $\mathcal{E}_0 \le \mathcal{E} \le \mathcal{E}_0 + \Delta\mathcal{E}$ can be obtained during a MC simulation from a history of duration T

$$\overline{v}(\mathcal{E}_0) = \frac{1}{T}\int_0^T v[\mathbf{k}(t)] \times \big(\theta[\mathcal{E}(t) - \mathcal{E}_0] - \theta[\mathcal{E}(t) - \mathcal{E}_0 - \Delta\mathcal{E}]\big)\, dt, \quad (14)$$

where $\theta(\mathcal{E})$ denotes the step function and $\mathbf{k}(t)$ represents the electron wave vector during the flight. Note that the overall mobility is proportional to the sum of the mean velocities of all intervalls.

A very interesting behavior can be observed when comparing the mean velocities resulting from simulations with classical and Fermi-Dirac statistics. The upper plot of Figure 21 shows that when considering only phonon scattering the mean velocities coincide for both simulation modes in the non-degenerate regime ($E_f - E_0 \approx -0.13$ eV). At high inversion layer concentrations, where the 2DEG is highly degenerate ($E_f - E_0 \approx 0.8$ eV), a shift of the mean velocity distribution toward higher energies and a decrease of its peak can be observed as compared to the mean velocity resulting from simulations without the Pauli principle. The coincidence of the mean velocities in the non-degenerate regime is merely a test that the algorithm with the Pauli principle included converges to the classical algorithm for the non-degenerate 2DEG. At high inversion layer concentrations

the different mean velocities can be interpreted as follows: In simulations neglecting the Pauli principle electrons will have an equilibrium energy of $k_B T$ whereas the mean energy resulting from simulations with the Pauli principle can be more than twice as much. Since phonon scattering is merely proportional to the density of states, which is an increasing step-like function in the 2DEG, electrons being at higher energies – as it is the case in simulations with the Pauli principle – experience more scattering and thus the phonon-limited mobility is strongly decreased (see Figure 22).

The lower plot of Figure 21 shows the mean velocities when surface roughness scattering has been included in simulations. In the low inversion layer density regime (depicted in grey) surface roughness scattering does not play an important role, and the mean velocities compare well with the simulation results with only phonon scattering included. Interestingly now even at high inversion layer densities the mean velocities (depicted in black) stemming from simulations with and without the Pauli principle do not differ as much. The large peak in the mean velocity at small energies that could be observed in simulations including only phonon scattering but neglecting the Pauli principle is here suppressed. This stems from the influence of surface roughness scattering which is more effective at small energies in contrast to phonon scattering.

Finally from Figure 21 one can also observe that due to degeneracy effects electrons at energies below the Fermi level have smaller velocities which corresponds to the general picture that these electrons have little contribution to transport.

The new algorithm is used to extract the effective mobility in unstrained and biaxially tensile strained Si inversion layers. For the simulations a one-dimensional Schrödinger-Poisson solver [48] is used with modifications to account for the energy splitting between the twofold and the fourfold conduction-band minima and for the change of the band-gap [41]. From these results the matrix elements for phonon and surface roughness scattering and the form factors are calculated following [49]. Screening of the surface roughness scattering was included according to [49] while impurity scattering has been ignored as we mainly focus on the high-density (high effective field) region. The dielectric function was treated as a tensor quantity and not as a scalar function as the latter approximation is only valid for ideal 2D systems where the wave functions have a δ-like shape. Furthermore the plasma dispersion function was not used to calculate the polarization function, as the plasma dispersion function underestimates the polarization function in the degenerate case. A non-parabolic bandstructure ($\alpha = 0.5$ eV^{-1}) was used leading to a 20% reduction of the phonon-limited mobility at 300 K in agreement with [41]. For all simulations presented here a uniform doping concentration of 2×10^{16} cm^{-3} has been assumed.

The simulated mobility curve for unstrained Si in the upper plot of Figure 22 shows good agreement with the universal mobility curve by Takagi [50]. Surprisingly, the effective mobility resulting from simulations with degenerate statistics are in close agreement to those using classical statistics even though the phonon-limited mobility experiences a noticeable reduction when using degenerate statistics. As previously discussed, this close agreement can only be understood from the cancellation of two effects: Degeneracy leads to an increase of the mean kinetic energy. This leads to an increase in phonon scattering and a decrease in the mobility. At the same time electrons with larger kinetic energies experience less effective surface-roughness scattering, thus the surface roughness limited mobility is increased. In unstrained Si by chance these two effects cancel each other at all effective fields, and the difference between a simulation with non-degenerate and degenerate statistics is very small.

Having calibrated our model against the unstrained universal mobility curve, a simulation of biaxially strained Si on relaxed $Si_{1-y}Ge_y$ substrates with $y = 0.25$ was performed. From Figure 22 it can be observed that in SSi inversion layers, where the ratio between phonon and surface roughness scattering is different – due to suppressed intervalley transitions – simulations with degeneracy effects yield higher mobilities $\mu_{\mathrm{sr,deg}} > \mu_{\mathrm{sr,nondeg}}$.

Our simulation results for the inversion layer mobility in biaxially strained Si on $Si_{1-y}Ge_y$ for various Ge contents suggest a saturation of the mobility enhancement at $y \approx 25\%$. As can be seen from Figure 22 the anomalous intersection of the strained and unstrained mobility curve from [41] was not observed, however Figure 23 indicates that the predicted mobility for SSi is still underestimated.

4.4 Conclusion

By means of MC simulations we are able to deduce the effect of degeneracy both on the phonon-limited mobility and the effective mobility including surface-roughness scattering. It is shown that in the unstrained case the inclusion of the Pauli principle leads to a noticeable reduction of the phonon-limited mobility, but has almost no impact on the effective mobility. The effective mobility of strained inversion layers increases slightly at high inversion layer concentrations when taking into account degenerate statistics. Thus a correct treatment of degenerate carrier statistics of the 2DEG of strained Si inversion layers is important.

However, this study cannot explain the experimental mobility enhancement for SSi, which is still underestimated at large effective fields, where surface roughness scattering dominates. Thus a careful revision of surface rough-

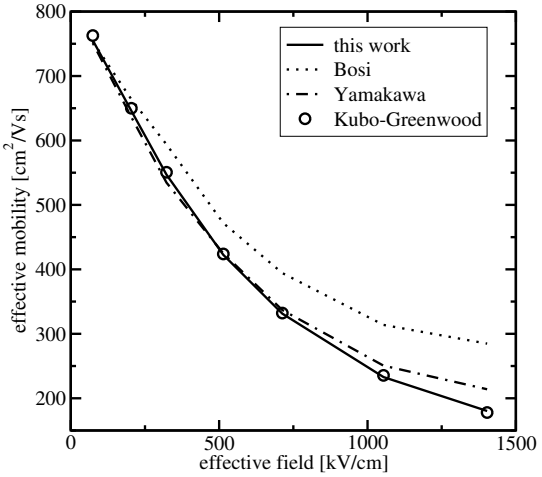

Figure 20: The simulated effective mobility using the new algorithm (solid line) is compared to results of a non-selfconsistent version of Bosi's algorithm [44] (dotted line), the algorithm proposed by [45], and to the mobility calculated with the Kubo-Greenwood formalism (open circles).

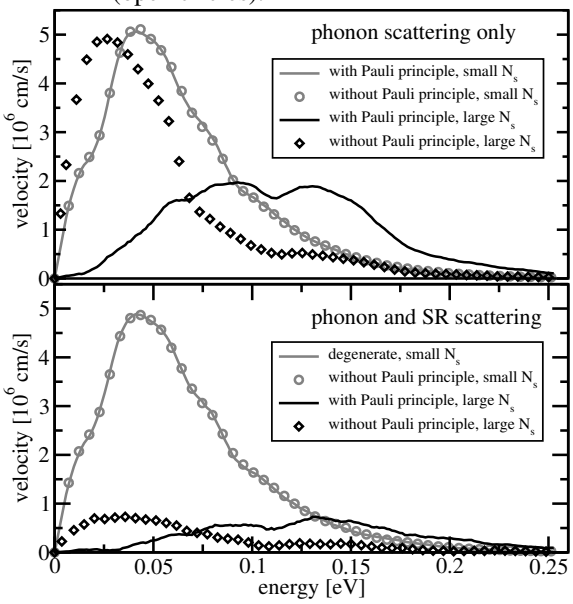

Figure 21: Mean velocities resulting from simulation with (lines) and without (symbols) inclusion of the Pauli principle for an inversion layer density $N_s \approx 10^{11}$ cm^{-2} (grey) and $N_s \approx 1.5 \times 10^{13}$ cm^{-2} (black).

ness scattering might be needed to achieve the correct mobility enhancements.

4.5 Acknowledgement

This work has been partly supported by the European Commission, project SINANO, IST 506844.

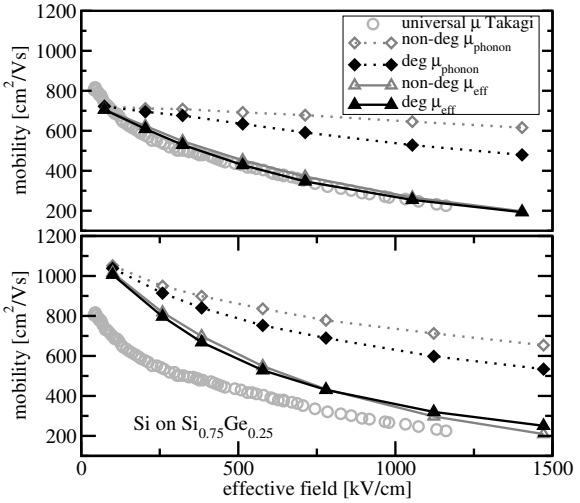

Figure 22: Calculated phonon-limited and effective mobility compared to the universal mobility curve with and without degeneracy effects for unstrained Si (upper curve) and biaxially strained Si (lower curve).

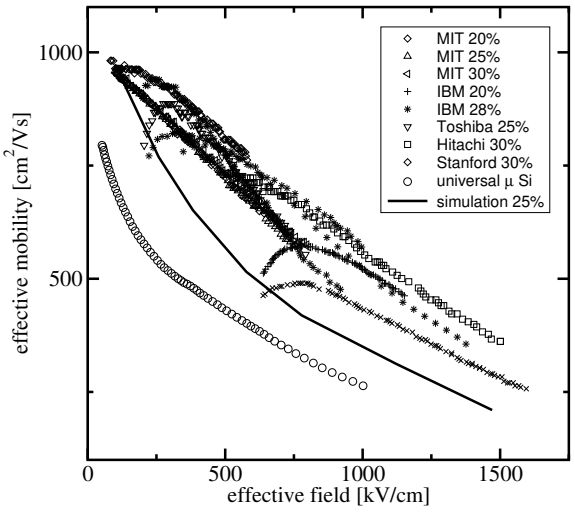

Figure 23: Comparison of simulations results for the effective mobility of biaxially strained Si on $Si_{1-y}Ge_y$ with $y = 0.25$ with published data from different experimental groups for various substrate compositions of Ge.

5 Quantum Correction for DG MOSFETs

The characteristics of modern semiconductor devices are strongly influenced by quantum mechanical effects. Due to this fact, purely classical device simulation is not sufficient to accurately reproduce the device behavior. For instance, the classical semiconductor equations have to be adapted to account for the quantum mechanical decrease of the carrier concentration near the gate oxide. Several available quantum correction models are derived for devices with one single inversion layer and are therefore only of limited use for thin double gate (DG) MOSFETs where the two inversion layers interact. We present a highly accurate quantum correction model which is even valid for extremely scaled DG MOSFET devices. Our quantum correction model is physically based on the bound states that form in the Si film. The eigenenergies and expansion coefficients of the wave functions are tabulated for arbitrary parabolic approximations of the potential in the quantum well. Highly efficient simulation of DG MOSFET devices scaled in the decananometer regime in TCAD applications is made possible by this model.

5.1 Introduction

Continously downscaling of the device geometry increases the influence of quantum mechanical effects on the device characteristics. Besides tunneling, the effect of quantum confinement highly affects the characteristics of bulk, silicon-on-insulator (SOI), and double gate (DG) MOSFET devices under inversion conditions.

Purely classical device simulation without adequate quantum correction is no longer sufficient to provide proper results since it predicts an exponential increase of the carrier concentration towards the gate oxide interface. However, due to quantum confinement, which affects the local density of states, the carrier concentration near the gate oxide decreases as shown in Figure 24.

Schrödinger Poisson (SP) solvers, which deliver a self consistent solution of a quantum mechanically calculated carrier concentration and the Poisson equation, provide accurate results for the carrier concentration within 1d slices perpendicular to the oxide interface. However, since the evaluation of the quantum mechanical electron density is computationally very demanding, the application of SP solvers is impractical.

In order to obtain proper results at significantly reduced CPU time, several quantum correction models for classical simulations have been proposed [51, 52, 53, 54, 55]. However, some of these corrections are based on empirical fits with numerous parameters [52, 53]. In some other models, the dependence on the electrical field adversely affects the convergence behavior [51]. Practically, the model proposed in [54] has to be recalibrated for each device. A comprehensive comparison of these models can be found in [56]. In addition, none of these models is suitable for highly scaled DG MOSFETs in the deca nanometer regime where two coupled inversion regions occur. In this work, we present a new, physically based, and more specific approach for state-of-the-art DG MOSFETs.

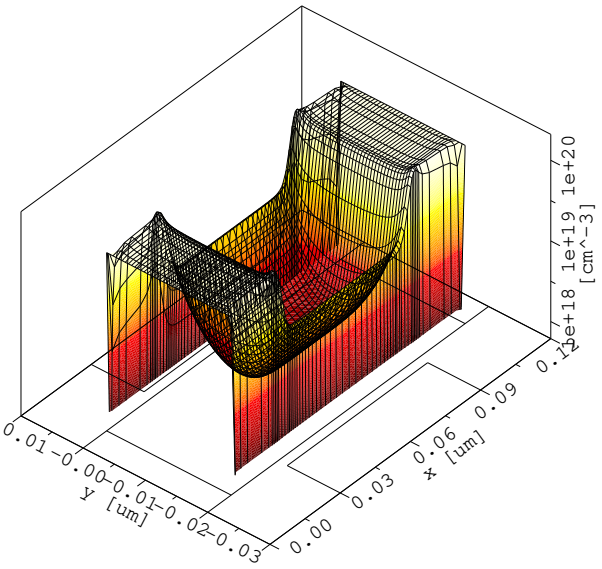

Figure 24: Quantum mechanical electron concentration within a DG MOSFET device. The concentration strongly decreases towards the oxide interfaces due to geometrical confinement.

5.2 Approach

The classical carrier concentration is based on the assumption of a free 3-dimensional electron gas and Boltzmann statistics and shows exponential increase towards the semiconductor-oxide interface. However, the physically correct quantum mechanically derived carrier concentration strongly decreases towards the interface. The classically derived concentration is adjusted to be equal to the quantum mechanically calculated carrier concentration [57] by introducing the quantum correction potential φ_{corr} as

$$n_{\mathrm{cl,corr}} = N_{\mathrm{C}} \exp\Big(- \frac{\mathcal{E}_{\mathrm{c}} - q\varphi_{\mathrm{corr}} - \mathcal{E}_{\mathrm{F}}}{\mathrm{k_B}T} \Big),$$

$$n_{\mathrm{qm}} = N_{\mathrm{C1}} \sum_n |\Psi_n(x)|^2 \exp\Big(- \frac{E_n - \mathcal{E}_{\mathrm{F}}}{\mathrm{k_B}T} \Big).$$

Here, N_{C} and N_{C1} denote the effective density of states for classical and the quantum mechanical carrier concentration, respectively. φ_{corr} describes the quantum correction potential, $\mathcal{E}_{\mathrm{c}}$ the conduction band edge energy, and $\mathcal{E}_{\mathrm{F}}$ the Fermi energy.

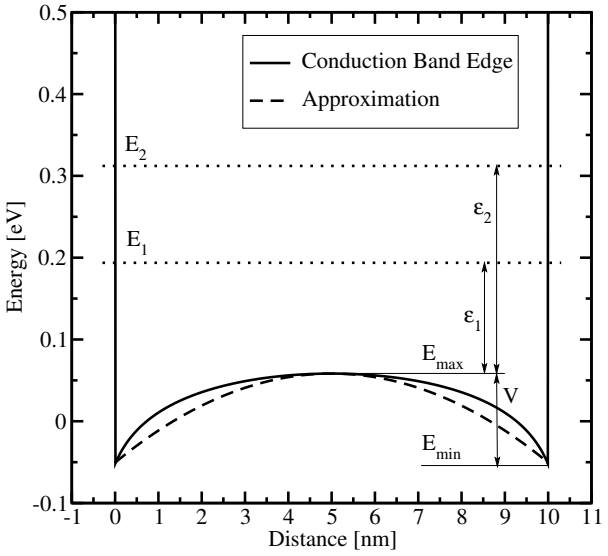

Figure 25: A cut of the conduction band edge energy perpendicular to the semiconductor-oxide interfaces. The dashed line displays its parabolic approximation. Furthermore, Eigenenergy levels and their relative values to $E_{\max}$ and $E_{\min}$ are shown.

This approach requires the knowledge of the energy levels E_n and the wavefunctions $\Psi_n(x)$ of the quantized states. To avoid the computationally expensive solution of the Schrödinger equation, we tabulate the solutions for a parabolic shaped approximation of the conduction band edge,

$$\mathcal{E}_{\mathrm{c}}(x) = E_{\max} - a(d/2 - x)^2\,,$$

as displayed in Figure 25. Input parameters are the film thickness d and the curvature a which is derived from an initial classical simulation. The wave functions are expanded as

$$\Psi_n(x) = \sum_k \xi_{n,k} \sqrt{\frac{2}{d}} \sin\left(\frac{\pi}{d} k x\right).$$

Hence, the offset of the energy levels ϵ_n and the expansion coefficients of the wavefunctions $\xi_{n,k}$ can be found by interpolation of tabulated values. This allows one to estimate a correction potential φ_{corr} such that the corrected classical carrier concentration is consistent with the SP solution

$$\exp\Big(-\frac{q\varphi_{\mathrm{corr}}}{\mathrm{k_B}T}\Big) = \exp\Big(-\frac{a(d/2-x)^2}{\mathrm{k_B}T}\Big) \times \sum_m \frac{N_{\mathrm{C1},m}}{N_{\mathrm{C}}} \sum_n |\Psi_{m,n}(x)|^2 \exp\Big(-\frac{\epsilon_{m,n} - \mathcal{E}_{\mathrm{F}}}{\mathrm{k_B}T}\Big).$$

Here, m denotes the summation over the different valley sorts (three for silicon) [58].

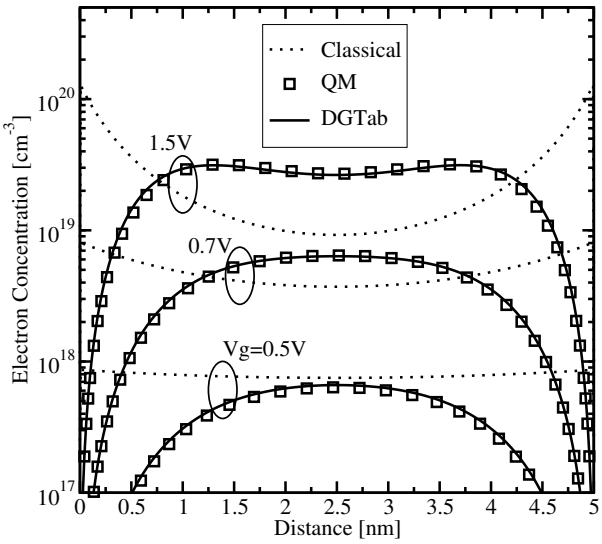

Figure 26: The classical, the quantum mechanical, and the corrected classical electron concentration of a double gate MOSFET structure with 5 nm silicon film thickness. Gate voltages of 0.5 V, 0.7 V, and 1.5 V were applied. DGTab quantum corrected curves show outstanding agreement with quantum mechanically derived curves for all applied gate voltages.

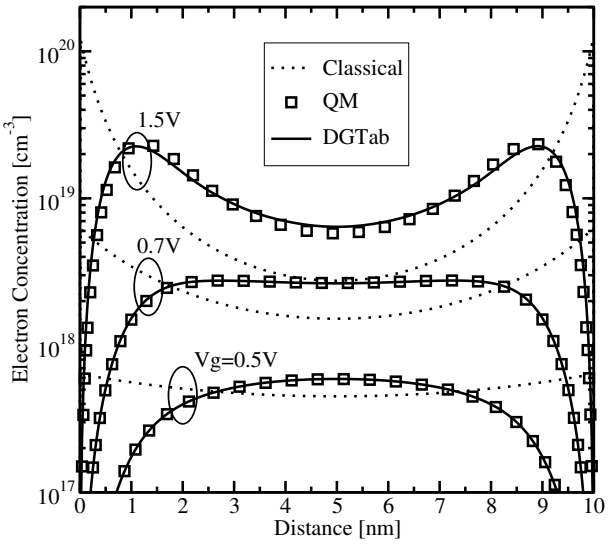

Figure 27: The classical, the quantum mechanical, and the corrected classical electron concentration of a double gate MOSFET structure with 10 nm silicon film thickness at gate voltages of 0.5 V, 0.7 V, and 1.5 V respectively. DGTab quantum corrected curves show outstanding agreement with quantum mechanically derived curves for all applied gate voltages.

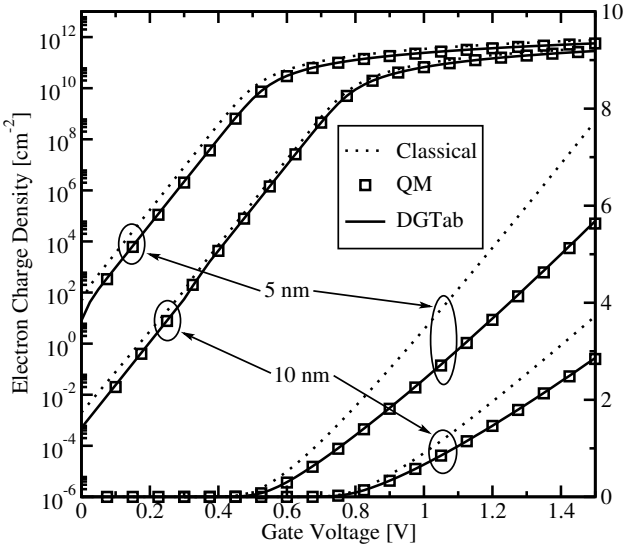

Figure 28: The total amount of the inversion charge as a function of the gate voltage for DG MOSFETs with film thicknesses of 5 nm and 10 nm respectively. The upper left curves are plotted in logarithmic scale, the lower right ones in linear scale. DGTab quantum corrected curves show outstanding agreement with quantum mechanically derived curves in the entire range of gate voltages.

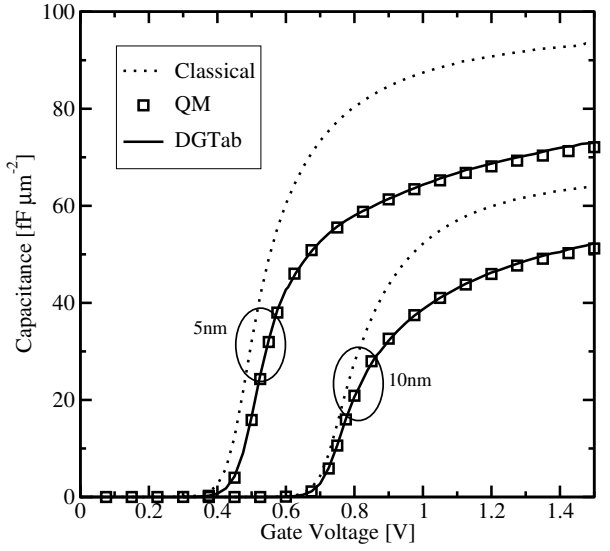

Figure 29: The resulting gate capacitance as a function of the gate voltage for DG MOSFETs with film thicknesses of 5 nm and 10 nm, respectively. DGTab quantum corrected curves show outstanding agreement with quantum mechanically derived curves for both film thicknesses.

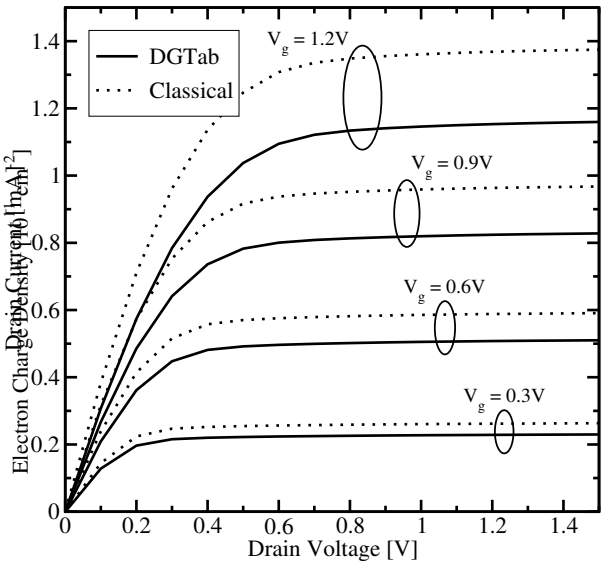

Figure 30: Output characteristics of a double gate MOSFET device with a Si film thickness of $20\,\mathrm{nm}$. The current is overestimated by a purely classical simulation because of the exponentially increasing carrier concentration near the oxide interfaces.

5.3 Results

We implemented this model in our general purpose device simulator MINIMOS-NT [5]. Our SP simulator VSP was used to derive the reference QM curves. The applied iteration scheme of an initial classical simulation followed by a single quantum correction step and a final classical simulation with corrected bandedges delivers results in the same accuracy as a complete self consistent simulation.

Figure 26 and Figure 27 show the electron concentration at different bias points for DG MOSFETs with $5\,\mathrm{nm}$ and $10\,\mathrm{nm}$ film thickness. Outstanding agreement between the QM and the corrected classical curves (DGTab) is achieved.

Both the inversion charge and the gate capacitance shown in Figure 28 and Figure 29 demonstrate excellent agreement between DGTab and S/P curves for a wide range of gate voltages and relevant film thicknesses. Since the derived inversion charge is based on the accurate carrier concentration, no further fitting parameters had to be introduced.

Figure 30 displays the output characteristics of a double gate MOSFET with a silicon film thickness of $20\,\mathrm{nm}$. Purely classical simulation overestimates the current in the entire range because of the increased carrier concentration near the oxide interfaces.

5.4 Conclusion

We derived a quantum correction model specifically for double gate MOSFETs based on the bound states that form in the silicon film. The model accurately reproduces both the carrier concentration distribution and gate capacitance characteristics as well as the total inversion charge even for extremely scaled DG MOSFET devices. Due to its computational efficiency, the model is well suited for TCAD simulation environments.

5.5 Acknowledgement

This work has been partly supported by the European Commission, project SINANO, IST 506844 and from the Austrian Science Fund, special research project IR-ON (F2509-N08).

References

[1] B. Kaczer, V. Arkhipov, R. Degraeve, N. Collaert, G. Groeseneken, and M. Goodwin. Temperature Dependence of the Negative Bias Temperature Instability in the Framework of Dispersive Transport. *Appl.Phys.Lett.*, 86:143506, 2005.

[2] M. Ershov, S. Saxena, H. Karbasi, S. Winters, S. Minehane, J. Babcock, R. Lindley, P. Clifton, M. Redford, and A. Shibkov. Dynamic Recovery of Negative Bias Temperature Instability in p-Type Metal-Oxide-Semiconductor Field-Effect Transistors. *Appl.Phys.Lett.*, 83(8):1647–1649, 2003.

[3] K. O. Jeppson and C. M. Svensson. Negative Bias Stress of MOS Devices at High Electric Fields and Degradation of MNOS Devices. *J.Appl.Phys.*, 48(5):2004–2014, 1977.

[4] S. Ogawa and N. Shiono. Generalized Diffusion-Reaction Model for the Low-Field Charge Build Up Instability at the Si/SiO_2 Interface. *Phys.Rev.B*, 51(7):4218–4230, 1995.

[5] Institut für Mikroelektronik, Technische Universität Wien, Austria. *MINIMOS-NT 2.1 User's Guide*, 2004. `www.iue.tuwien.ac.at/software/`.

[6] M. A. Alam and S. Mahapatra. A Comprehensive Model of PMOS NBTI Degradation. *Microelectron.Reliab.*, 45(71-81):71–81, 2005.

[7] A. Haggag, W. McMahon, K. Hess, K. Cheng, J. Lee, and J. Lyding. High-Performance Chip Reliability from Short-Time-Tests. In *Proc. Intl.Rel.Phys.Symp.*, pp 271–279, 2001.

[8] G. Dos Reis and J. Jarvi. What is Generic Programming? In *Library Centric Sofware Design, OOPSLA*, San Diego, CA, USA, 2005.

[9] G. Berti. *Generic Software Components for Scientific Computing*. PhD thesis, Technische Universität Cottbus, 2000.

[10] J. Siek, L.-Q. Lee, and A. Lumsdaine. *The Boost Graph Library: User Guide and Reference Manual*. Addison-Wesley, 2002.

[11] D. Abrahams, J. Siek, and T. Witt. New Iterator Concepts. Technical Report N1477 03-0060, ISO/IEC JTC 1, Information Technology, Subcommittee SC 22, Programming Language C++, 2003.

[12] D. Gregor, J. Willcock, and A. Lumsdaine. Concepts for the C++0x Standard Library: Iterators. Technical Report N2039=06-0109, ISO/IEC JTC 1, Information Technology, Subcommittee SC 22, Programming Language C++, 2006.

[13] Boost. *Boost Lambda Library*. `http://www.boost.org`.

[14] Boost. *Boost Phoenix*, 2004. `http://spirit.sourceforge.net/`.

[15] M. Spevak, R. Heinzl, P. Schwaha, T. Grasser, and S. Selberherr. A Generic Discretization Library. In *Library Centric Sofware Design, OOPSLA*, Portland, OR, USA, 2006.

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

[17] S. Selberherr, A. Schütz, and H.W. Pötzl. MINIMOS—A Two-Dimensional MOS Transistor Analyzer. *IEEE Trans. Electron Dev.*, ED-27(8):1540–1550, 1980.

[18] S. Halama, Ch. Pichler, G. Rieger, G. Schrom, T. Simlinger, and S. Selberherr. VISTA — User Interface, Task Level, and Tool Integration. *IEEE J.Techn. Comp. Aided Design*, 14(10):1208–1222, 1995.

[19] R. Sabelka and S. Selberherr. A Finite Element Simulator for Three-Dimensional Analysis of Interconnect Structures. *Microelectronics Journal*, 32(2):163–171, 2001.

[20] T. Binder, A. Hössinger, and S. Selberherr. Rigorous Integration of Semiconductor Process and Device Simulators. *IEEE Trans.Comp.-Aided Design of Int. Circ. and Systems*, 22(9):1204–1214, 2003.

[21] W. Benger. *Visualization of General Relativistic Tensor Fields via a Fiber Bundle Data Model*. PhD thesis, Freie Universität Berlin, 2004.

[22] P. A. Markowich, C.A. Ringhofer, and C. Schmeiser. *Semiconductor Equations*. Springer, Wien-New York, 1990.

[23] A. Sheikholeslami, E. Al-Ani, R. Heinzl, C. Heitzinger, F. Parhami, F. Badrieh, H. Puchner, T. Grasser, and S. Selberherr. Level Set Method Based General Topography Simulator and its Applications in Interconnect Processes. In *Intl. Conf. on Ultimate Integration of Silicon*, pp 139–142, Bologna, Italy, 2005.

[24] K. Jänich. *Topologie*. Springer, Heidelberg, 2001.

[25] A. J. Zomorodian. Topology for Computing. In *Cambridge Monographs on Applied and Computational Mathematics*, 2005.

[26] J. R. Shewchuk. *Delaunay Refinement Mesh Generation*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1997.

[27] J. Siek and A. Lumsdaine. Mayfly: A Pattern for Lightweight Generic Interfaces. In *Pattern Languages of Programs*, 1999.

[28] M. H. Austern. *Generic Programming and the STL: Using and Extending the C++ Standard Template Library*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1998.

[29] M. Zalewski and S. Schupp. Changing Iterators with Confidence. A Case Study of Change Impact Analysis Applied to Conceptual Specifications. In *Library Centric Sofware Design, OOPSLA*, San Diego, CA, USA, 2005.

[30] A. Fabri. CGAL- The Computational Geometry Algorithm Library, 2001. citeseer.ist.psu.edu/fabri01cgal.html.

[31] J. Litsios, B. Schmithüsen, U. Krumbein, A. Schenk, E. Lyumkis, B. Polsky, and W. Fichtner. DESSIS 3.0: Manual. Technical report, ISE Integrated Systems Engineering, Zürich, Switzerland, 1996.

[32] R. Sabelka and S. Selberherr. A Finite Element Simulator for Three-Dimensional Analysis of Interconnect Structures. *Microelectronics Journal*, 32:163–171, Jan. 2001.

[33] J.L. Volakis, A. Chatterjee, and L.C. Kempel. *Finite Element Method for Electromagnetics*. IEEE Press, New York, 1998.

[34] P. Fleischmann, R. Sabelka, A. Stach, R. Strasser, and S. Selberherr. Grid Generation for Three-dimensional Process and Device Simulation. In *SISPAD*, pp 161–166, 1996.

[35] C. Harlander, R. Sabelka, and S. Selberherr. Three-Dimensional Electro-Thermal Simulation of Interconnect Structures with Temperature-Dependent Permittivity. *InterPACK*, pp 1–2, 2001.

[36] C. Harlander, R. Sabelka, R. Minixhofer, and S. Selberherr. Three-Dimensional Transient Electro-Thermal Simulation. In *Proceedings THERMINIC Workshop*, pp 169–172, 1999.

[37] L.-J. Huang, J. Chu, S. Goma, C. Emic, S. Koester, D. Canaperi, P. Mooney, S. Cordes, J. Speidell, R. Anderson, and H. Wong. Carrier mobility enhancement in strained Si-on-insulator fabricatedby wafer bonding. *VLSI Symp. Tech.Dig.*, pp 57–58, 2001.

[38] K. Rim, J. Chu, H. Chen, K. Jenkins, T. Kanarsky, K. Lee, A. Mocuta, H. Zhu, R. Roy, J. Newbury, J. Ott, K. Petrarca, P. Mooney, D. Lacey, S. Koester, K. Chan, D. Boyd, M. Ieong, and H. Wong. Characteristics and device design of sub-100 nm strained Si N- and PMOSFETs. *VLSI Symp. Tech.Dig.*, pp 98–99, 2002.

[39] N. Sugii, D. Hisamoto, K. Washio, N. Yokoyama, and S. Kimura. Enhanced performance of strained-Si MOSFETs on CMP SiGe virtual substrate. *Intl.Electron Devices Meeting*, pp 737–740, 2001.

[40] S.-E. Thompson, M. Armstrong, C. Auth, M. Alavi, and M. Buehler. A 90-nm logic technology featuring strained-silicon. *IEEE Trans.Electron Devices*, 51:1790, 2004.

[41] M. V. Fischetti, F. Gamiz, and W. Hänsch. On the enhanced electron mobility in strained-silicon inversion layers. *J.Appl.Phys*, 92:7320, 2002.

[42] J. Watling, L. Yang, M. Borici, R. C. Wilkins, A. Asenov, J. R. Barker, and S. Roy. The impact of interface roughness scattering and degeneracy in relaxed and strained Si n-channel MOSFETs. *Solid-State Electron*, 48:1337, 2004.

[43] T. Ando, A. Fowler, and F. Stern. Electronic properties of two-dimensional systems. *Review of Modern Physics*, 54:437, 1982.

[44] S. Bosi and C. Jacoboni. Monte Carlo high-field transport in degenerate GaAs. *J.Physics C*, 9:315, 1976.

[45] S. Yamakawa, H. Ueno, K. Taniguchi, C. Hamaguchi, K. Miyatsuji, K. Masaki, and U. Ravaioli. Study of interface roughness dependence of electron mobility in Si inversion layers using the Monte Carlo method. *J.Appl.Phys.*, 79:911, 1996.

[46] D. Roychoudhury and P. K. Basu. Mobility of electrons in a quantized silicon inversion layer due to phonon scattering. *Physical Review B*, 22:6325, 1980.

[47] M. V. Fischetti and Z. Ren. Six-band k.p calculation of the hole mobility in silicon inversion layers: Dependence on surface orientation, strain, and silicon thickness,. *J.Appl.Phys.*, 94:1079, 2003.

[48] D. Vasileska and Z. Ren. SCHRED 2.0 User's Manual, 2000. `http://www.nanohub.org`.

[49] C. Jungemann, A. Edmunds, and W. L. Engl. Simulation of Linear and Nonlinear Electron Transport in Homogenous Silicon Inversion Layers. *Solid-State Electron.*, 36(11):1529–1540, 1993.

[50] S. Takagi, A. Toriumi, M. Iwase, and H. Tango. On the universality of inversion layer mobility in Si MOSFET's:Part I-effects of substrate impurity concentration. *IEEE Trans.Electron Devices*, 41:2357, 1994.

[51] M. Van Dort, P. Woerlee, and A. Walker. A Simple Model for Quantisation Effects in Heavily-Doped Silicon MOSFETs at Inversion Conditions. *Solid State Electronics*, 37(3):411, 1994.

[52] *Improved Modified Local Density Approximation for Modeling of Size Quantization in NMOSFETs*, 2001.

[53] *Modeling of Size Quantization in Strained Si-nMOSFETs with the Improved Modified Local Density Approximation*, volume 3, 2005.

[54] W. Hänsch, T. Vogelsang, R. Kircher, and M. Orlowski. Carrier Transport Near the Si/SiO2 Interface of a MOSFET. *Solid State Electronics*, 32:839, 1991.

[55] G. Paasch and H. Übensee. Carrier Density Near the Semiconductor-Insulator Interface – Local Density Approximation for Non-Isotropic Effective Mass. *Phys.Stat.Sol.*, 118(1):255, 1983.

[56] *Quantum Correction Models for Modern Semiconductor Devices*, volume 1, 2005.

[57] M. Karner, M. Wagner, T. Grasser, and H. Kosina. A Physically Based Quantum Correction Model for DG MOSFETs. In *Proc. MRS Spring Meeting 2006*, p 104, 2006.

[58] F. Stern. Self-Consistent Results for n-Type Si Inversion Layers. *Phys.Rev.B*, 5(12):4891–4899, 1972.