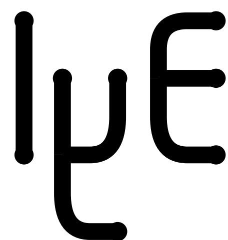
VISTA Status Report December 1995

M. Knaipp, Ch. Pichler, G. Rieger, M. Rottinger,
R. Sabelka, S. Selberherr, R. Strasser,



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

Contents

1 The VISTA Simulation Flow Control Module **1**

1.1 User Interface 1

1.2 Input Deck Templates 1

1.3 Simulation of Complex Processes 2

2 The PIF Editor as TCAD Data Processor **5**

2.1 Generation of a Diode 5

2.2 CCD Generation 5

3 Integration of SCAP into VISTA **8**

3.1 Introduction 8

3.2 Mathematical Background 8

3.3 Interface to PIF 8

3.4 Example 9

4 Transient Simulation Using MINIMOS-NT **12**

4.1 Features of the Transient Simulation 12

4.2 Time Dependent Contact Values 12

4.3 Time Step Estimation 12

4.4 Interpolation of Contact Values 14

4.5 Transient Simulation of a CCD 14

5 Modeling of Thermal Effects with MINIMOS-NT **18**

5.1 Differential Equations of the Lattice Heat Flow 18

5.2 Boundary Conditions 18

5.3 Thermal Energy Balance of a Device 19

5.4 Input Data for a Thermal Simulation 19

5.5 Example 20

1 The VISTA Simulation Flow Control Module

1.1 User Interface

Based on recent extensions of the VISTA Visual User Interface library (VUI), the user interface of the Simulation Flow Control module (SFC) has been integrated into the framework's main panel to provide direct access to the flow control and run data management facilities. Figure 1 shows the new VISTA shell with the system jobs and SFC subwindows.

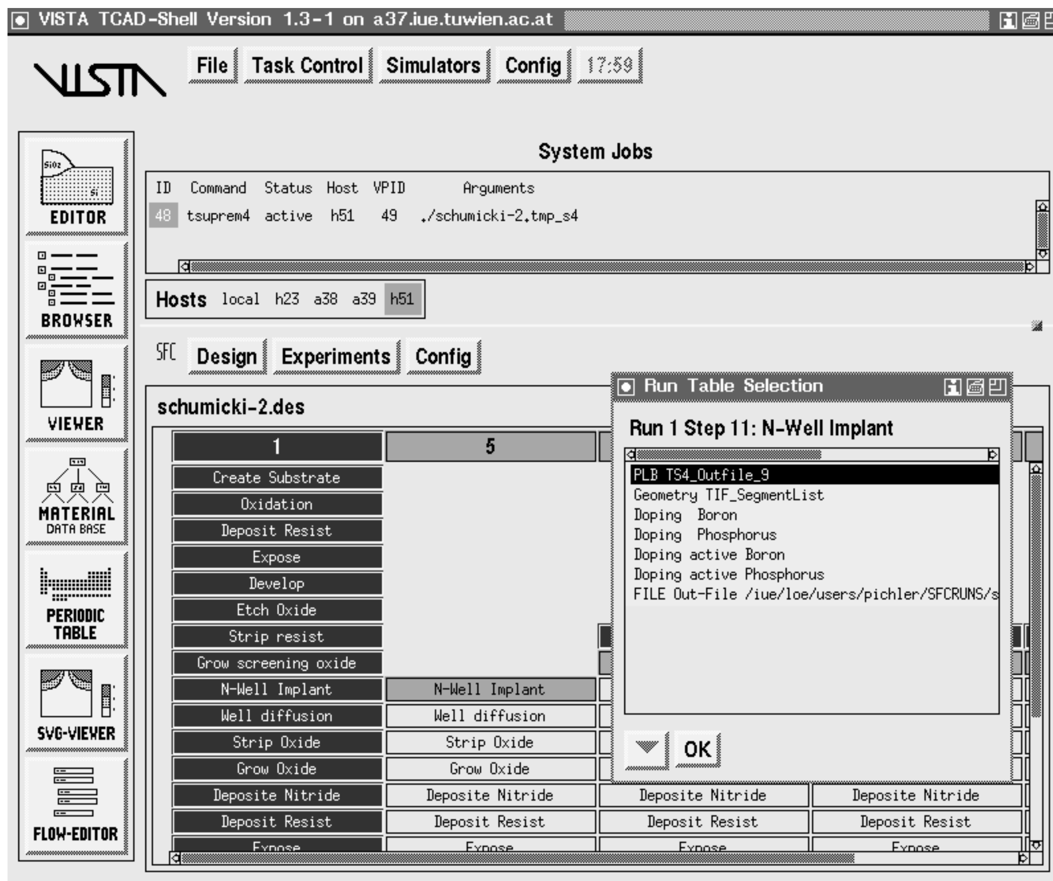


Figure 1: VISTA main shell with SFC interface

Frequently used utilities such as the PIF Editor, the PIF Browser, the material database, etc. are accessible directly on the left side of the interface. The SFC subwindow shows a number of runs on a spread sheet, with split branches being displayed to the right of the parent branch. A list selection provides direct access to all data generated by each step of the runs displayed. Postprocessors and viewers are started in accordance with the type of data to display. The system jobs display shows queued and active system jobs together with status information for all registered computing hosts.

1.2 Input Deck Templates

To facilitate the use of external simulators, also with existing input decks, a template-based input deck generator and an automatic user interface generator were developed. By marking numerical

or string entries in an input deck as parameters, existing simulator input files can be used in a simulation flow as separate steps. Parameter values can be displayed and edited with a graphical user interface automatically generated from the marks in the input file template.

Figure 2 shows a sequence from a simulator input file for a given simulation task. Figure 3 shows the template file generated from the input deck. By adding a symbolic name to a numerical or string expression, it is marked as parameter. Additional arguments such as `:text` or `:units` can be used to determine the appearance of the graphical user interface, to evaluate expressions before substituting values, and to format data. Figure 4 shows the user interface panel generated from the template file.

```
COMMENT NH IMP
 implant arsenic dose=3.5e15 energy=50

COMMENT  RTA ANNEAL
 diffusion temp=900 time=1 inert
 COMMENT  BPSG  ANNEAL
 diffusion temp=850 time=20 inert
```

Figure 2: input deck example

```
COMMENT NH IMP
 implant arsenic dose=<(nh-dose 3.5e15)> energy=<((nh-en :text "NH Energy") 50)>

COMMENT  RTA ANNEAL
 diffusion temp=900 time=<((rta-time :text "RTA Time" :units "min") 1)> inert
 COMMENT  BPSG  ANNEAL
 diffusion temp=850 time=20 inert
```

Figure 3: input deck template with parameter definitions

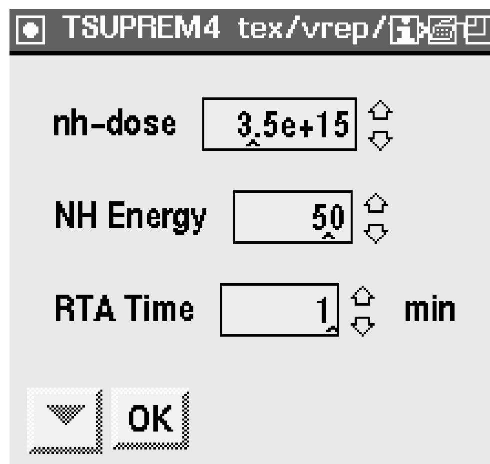


Figure 4: Panel for TSUPREM step generated automatically from input deck template.

1.3 Simulation of Complex Processes

The following example demonstrates the capability of the simulation flow control module. A complete CMOS process published in [1] is simulated using VISTA simulation tools as well as

Id	Process Step	Effect
1	Prepare Substrate	(100)-oriented, p-doped silicon ($\rho = 10\text{m}\Omega\text{cm}$) with a $12\mu\text{m}$ epi layer ($\rho = 10\Omega\text{cm}$)
2	N-Tub Implant	Phosphorus ($150\text{keV}, 5 \cdot 10^{12}\text{cm}^{-2}$)
3	N-Tub Diffusion	1150°, 150min, N_2
4	Field Oxide Mask	
5	Channel Stop Implant	Boron ($16\text{keV}, 5 \cdot 10^{13}\text{cm}^{-2}$)
6	Anti Punch Trough Implant	Boron ($180\text{keV}, 8 \cdot 10^{11}\text{cm}^{-2}$)
7	Field Oxidation	960°, $t_{ox} = 850nm$
8	Etchback	Reduce bird's beak by etching the oxide back to 500nm thickness
9	Gate Oxidation	$d_{ox} = 25\text{nm}$
10	Threshold Adjust Implant	Boron ($45\text{keV}, 9 \cdot 10^{11}\text{cm}^{-2}$)
11	Gate Formation	$d_{Poly} = 400\text{nm}$
12	N-LDD Implant	Phosphorus ($100\text{keV}, 2 \cdot 10^{13}\text{cm}^{-2}$)
13	Create LDD-Spacers	Deposit 300nm TEOS-oxide; anisotropic etch
14	N Source/Drain-Implant	Arsenic ($100\text{keV}, 5 \cdot 10^{15}\text{cm}^{-2}$)
15	P Source/Drain-Implant	Boron ($30\text{keV}, 3 \cdot 10^{15}\text{cm}^{-2}$)
16	Isolation Oxide	TEOS oxide ($d_{ox} = 800\text{nm}$)
17	Contact Etch	
18	Metal1 Deposition	Aluminum
19	Metal1 Etch	
20	Isolation Oxide	

Table 1: CMOS process recipe (simplified)

an external tool such as TSUPREM4. The given process basically contains all kinds of steps occurring in modern technologies. As far as topography (etching, deposition) is concerned the simulations are carried out by the PROMIS etch module. Lithography steps are performed by the SKETCH tool. Implantation, diffusion and oxidation are done with TSUPREM. The advantages of the physically based topography tools of VISTA are combined with the oxidation/diffusion capabilities of TSUPREM. An overview of the process is given in Table 1 (for details see [1], p.370).

The simulated structure is a CMOS inverter, containing most of the relevant intrinsic and parasitic devices. Figure 5 shows the concentration of phosphorus after the n-channel LDD implant. The final topography can be seen in Figure 6.

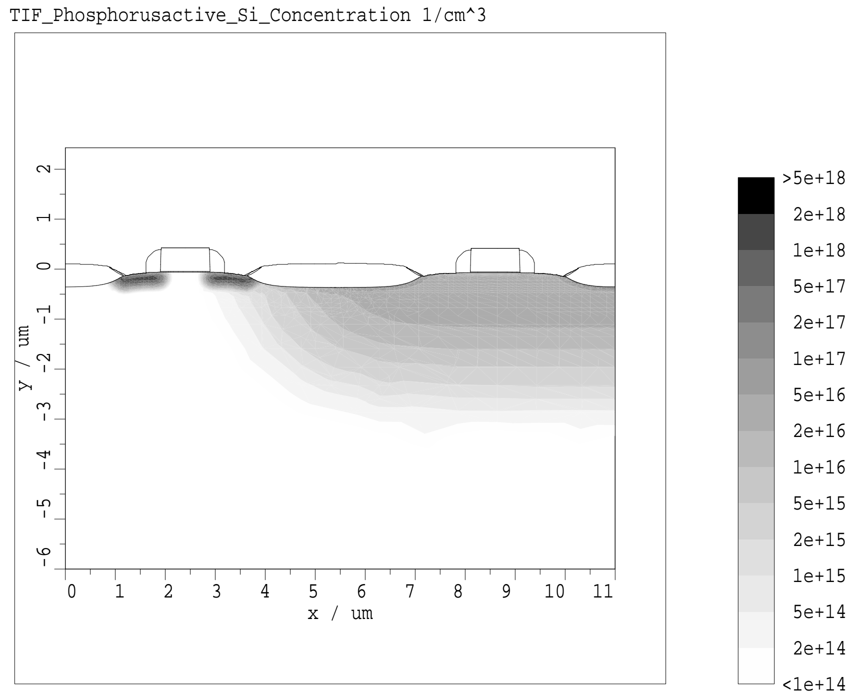


Figure 5: The distribution of phosphorus after the N-LDD implant

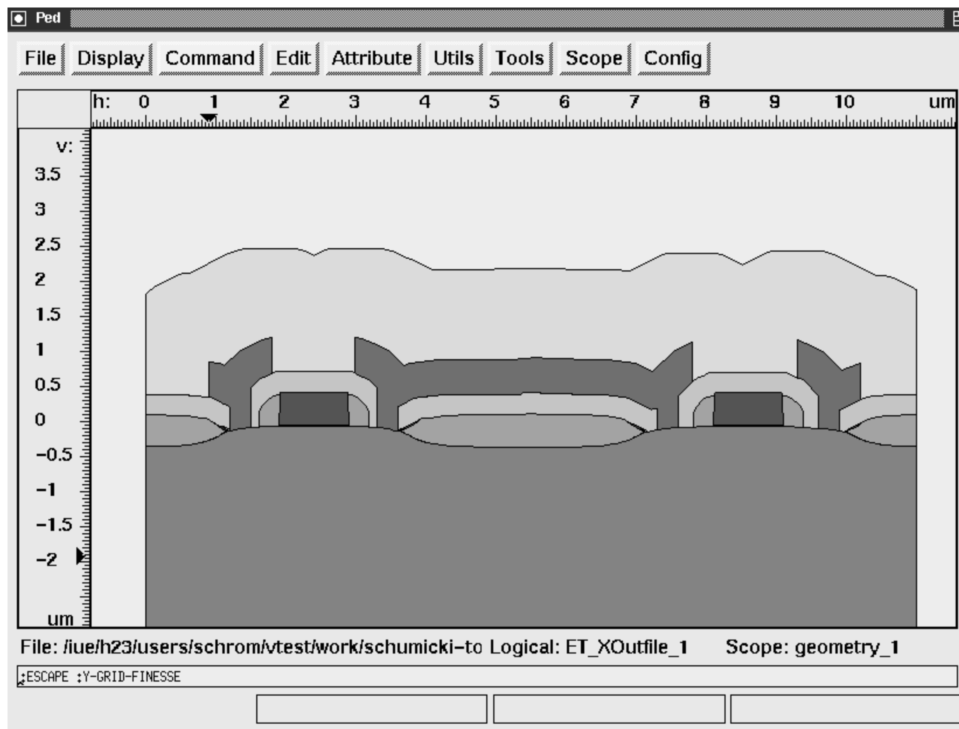


Figure 6: The final structure after passivation

2 The PIF Editor as TCAD Data Processor

With the integrated LISP interpreter the PIF Editor became a powerful and flexible tool for TCAD data manipulations [2, 3]. This induced desire to perform some of PED's operations not only interactively, but also in batch-mode, e.g., within the SFC. Therefore, a major redesign of PED took place, which enabled interactive and batch-mode operation, controlled either from the command line or from command files (which are supplied as small LISP programs).

One of the classic uses of PED has been the specification of devices for subsequent process or device simulations. While it has already been possible to write a LISP function that generates a device, the call to this function still required user interaction. Now, a transient call to the PIF Editor, with support of the function parameters on the command line, does the job. In the following, some example applications of the new features are demonstrated.

2.1 Generation of a Diode

As a simple but instructive example for the generation of a complete device the function `ped::diode` is introduced, which generates a diode structure from scratch. It consists of the following major steps:

- Creation of a 'vanilla' silicon block
- Generation of a grid (by default an ortho-grid generated, but the use of external grid generators - like TRIGEN - is also possible.)
- Generation of the phosphorus and boron dopings by simple analytical functions
- Creation of the insulating oxide and contacts with simple deposition functions

While the topology of the diode structure is hard-coded in the function, all geometric parameters as well as the implant positions, extents and peaks are scalable. `ped::diode` can be invoked from anywhere in the LISP code of PED: from a panel with interactive support of the parameters, from a LISP file, and from the command line.

2.2 CCD Generation

A more practical example is a charge-coupled device (CCD) which can be generated using a parameterized design. It is now possible to start PED, provide the parameters (doping levels, gate length, gate spacing, number of gates, etc.) to the function `ped::ccd` via a panel, view the resulting structure, and let the device be written to a PBF file directly. Figure 9 shows the geometry of a three-phase CCD, which was also simulated with MINIMOS-NT (see Section 4). The gate electrodes are automatically grouped into one segment per clock phase (indicated by different shadings).

```
(defun ped::diode
  (width depth hsplit vsplit plim blim ppeak bpeak bbury
   &key
   (left 0.0)                   ; left boundary of diode
   (contact-hole-width 0.5)
   (oxide1-thickness 1.0)
   (metal1-thickness 1.0)
   (bback 1e15)
   &aux x0 x1 y0 y1 s g)        ; some local variables

  (setq x0 (float left))
  (setq x1 (+ x0 width))
  (setq y0 (- (abs depth)))
  (setq y1 0.0)

  ; make a rectangle
  (setq s (rectangle (list x0 y0) (list x1 y1) :material "Si"))

  ; generate an ortho-grid
  (let
      (op ax dx ay dy i)
    (setq ax (make-array (1+ hsplit)))
    (setq dx (/ (- x1 x0) hsplit))
    (dotimes (i (1+ hsplit)) (setf (aref ax i) (* i dx)))
    (setq ay (make-array (1+ vsplit)))
    (setq dy (/ (- y1 y0) vsplit))
    (dotimes (i (1+ vsplit)) (setf (aref ay i) (* i dy)))
    (setq g (ortho  (list x0 y0) '((1 0) (0 1))  (list ax ay))))

  ; implant the dopings
  (when g
        (implant left plim 0.0 ppeak 5 5 :grid g :material "P")
        (implant (+ left blim) (- width blim) 0.0 bpeak
                 (sqrt 3.0) (sqrt 3.0) :grid g :material "B")
        (implant left width depth bbury 2 2 :grid g :material "B")
        (background-doping bback :grid g :material "B")
        )

  ; make the oxide
  (deposit (+ left (/ contact-hole-width 2.))  (- width contact-hole-width)
           oxide1-thickness  :material "SiO2")

  ; make left contact
  (deposit left (/ width 2.)
           metal1-thickness  :depomode :flat-min  :material "Al")

  ; make right contact
  (deposit (- (+ left width) (/ width 5.))  (/ width 5.)
           metal1-thickness  :depomode :flat-min  :material "Al")

  ; make metal1-oxide
  (deposit (+ left (/ width 2))  (* 0.3 width)
           metal1-thickness  :material "SiO2"))
```

Figure 7: LISP code for generating a diode structure

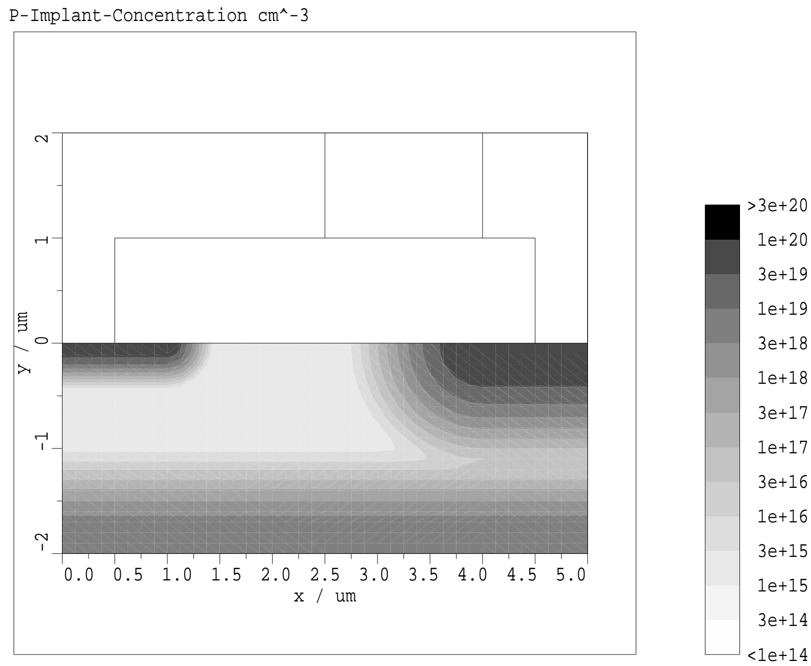


Figure 8: Net doping generated by function `ped::diode`

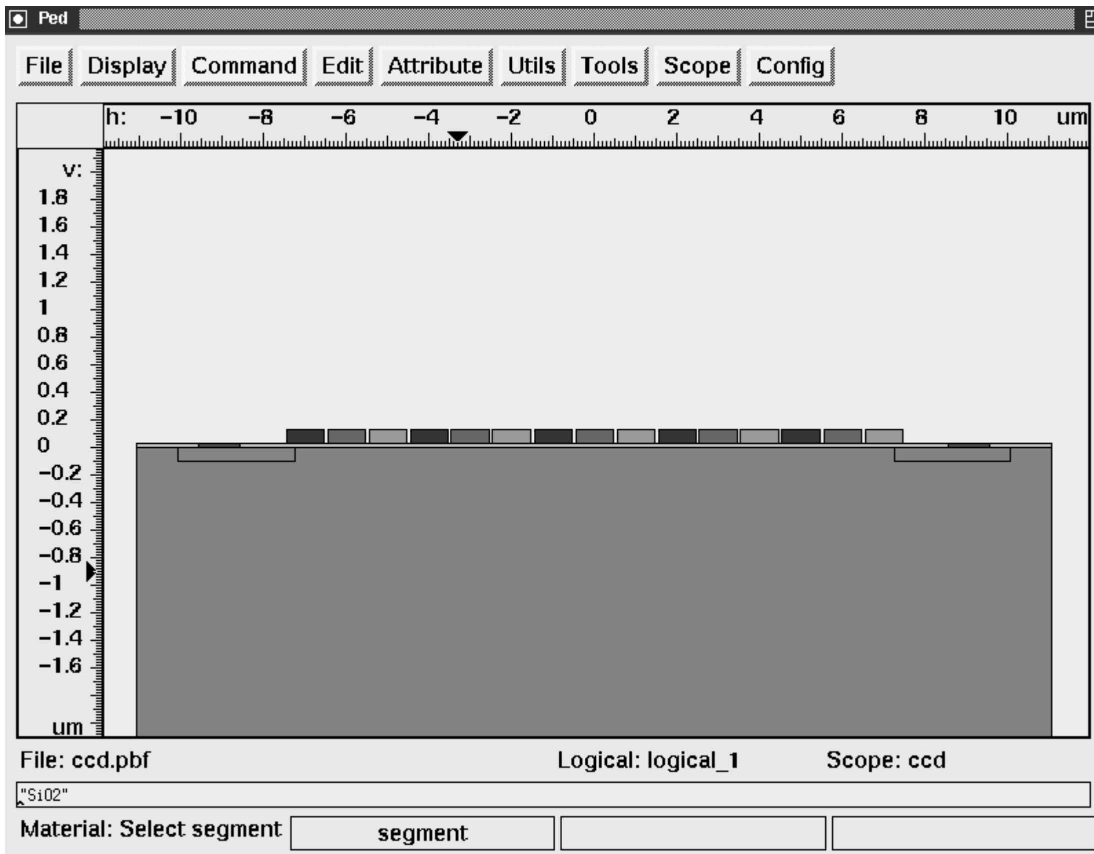


Figure 9: CCD structure generated with PED

3 Integration of SCAP into VISTA

The three-dimensional capacitance extraction program SCAP (Smart Capacitance Analysis Program [4]) has been integrated into VISTA. Therefore a flexible simulator for calculating linear capacitances in two- and three-dimensional structures is available and can communicate with other process simulation modules within VISTA. Furthermore, due to the integration the simulator profits from the large variety of already existing geometry, gridding and visualization tools of the VISTA framework.

3.1 Introduction

The simulator SCAP extracts linear capacitances of two and three dimensional wiring structures. It is assumed that the conductor material has infinite conductivity. The conductors represent Dirichlet boundary conditions for the electrostatic potential. The program uses the finite element method to calculate the electrostatic field energy for several applied conductor potentials, and the capacitances are extracted from the calculated energies. The energy method achieves a higher accuracy than the method of charge integration on the conductors. A preconditioned conjugate gradient method has been implemented to solve the large linear system in order to obtain the potential distribution and energy. To achieve an efficient utilization of computer memory, a compressed matrix format for the sparsely occupied stiffness matrix is implemented.

3.2 Mathematical Background

Field calculation method The variational formulation of the solution of the Laplace equation

$$\mathrm{div}\ \varepsilon(x,y,z)\ \mathrm{grad}\ \psi(x,y,z) = 0 \tag{1}$$

is the equivalent formulation of the minimization of the functional

$$I = \varepsilon_0 \int\limits_G \varepsilon_r(x,y,z) \left(\left(\frac{\partial \psi}{\partial x}\right)^2 + \left(\frac{\partial \psi}{\partial y}\right)^2 + \left(\frac{\partial \psi}{\partial z}\right)^2 \right) dV \quad I \rightarrow min \tag{2}$$

which holds exactly twice the electrostatic field energy E_{pot}.

Capacitance extraction method A charge balanced $n-$conductor system needs n calculated energy values to extract the $n(n-1)/2$ capacitance values. The common form

$$E_{pot} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij}\ \psi_i\ \psi_j \tag{3}$$

can be rewritten for linear dielectrics to

$$E_{pot} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij}\ (\psi_i - \psi_j)^2\ . \tag{4}$$

3.3 Interface to PIF

To read the input geometry and material properties and write back the calculated capacitance values SCAP has been equipped with a PIF interface.

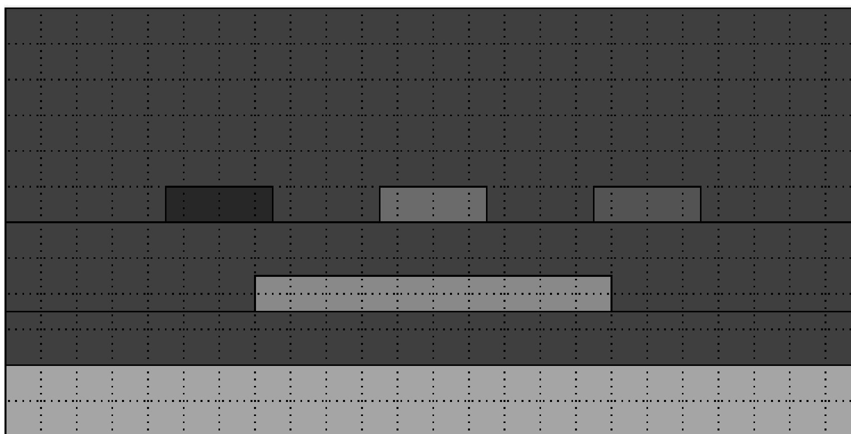


Figure 10: Five-conductor interconnect structure

Requirements for the input PIF file SCAP needs for its calculations the geometry of the problem, as well as information on some material properties of the individual segments of the input geometry. Therefore the input PIF file must contain exactly one geometry construct and a “SegmentDescription” which must contain the “MaterialType” of each segment of the geometry. Furthermore, “Resistivity”-, (or “Conductivity”-) and “Permittivity”-attributes may be specified. If not, SCAP gets this information from the VISTA material database. The resistivity (or conductivity) values are used for distinguishing between insulators and conductors by comparison with a configurable threshold. Both conductors and insulators are considered as ideal.

SCAP also expects a calculation grid to be found on the input PIF file. It must be conformal with the boundaries of the geometric segments and cover at least all insulators. The simulation grid can easily be generated during a preprocessing step with a grid generator available in the VISTA framework (e.g., TRIGEN). After reading the grid, SCAP eliminates all grid elements in conducting segments. Segments with common borders are considered as electrically joined and represent one logical Segment for the capacitance calculation. Dirichlet boundary conditions are assigned to grid points on the interface between conductors and insulators while all other points on the border of the geometry implicitly have homogeneous Neumann boundary conditions.

Writing the results After calculating the electric field distributions and deriving the partial capacitance matrix from the electric field energies, the capacitance values are written back to the PIF file into a “Contact2Description” attribute. A “ContactDescription” attribute is used to identify the conducting segments of the geometry, between which the capacitances are defined. Optionally the potential distributions of individual simulation runs can be written to PIF.

3.4 Example

The following example shows a simple two-dimensional five-conductor example. The geometry has been drawn with the PED (PIF editor). It consists of a ground plane and two metal layers (see Figure 10). SiO_2 is used as dielectric.

The simulation grid has been generated using the TRIGEN grid generator (see Figure 11). The calculated capacitance values are presented in Table 2. The capacitance indices refer to the segment numbers on the PIF file. (The ground plane has the index 1, the conductor above has

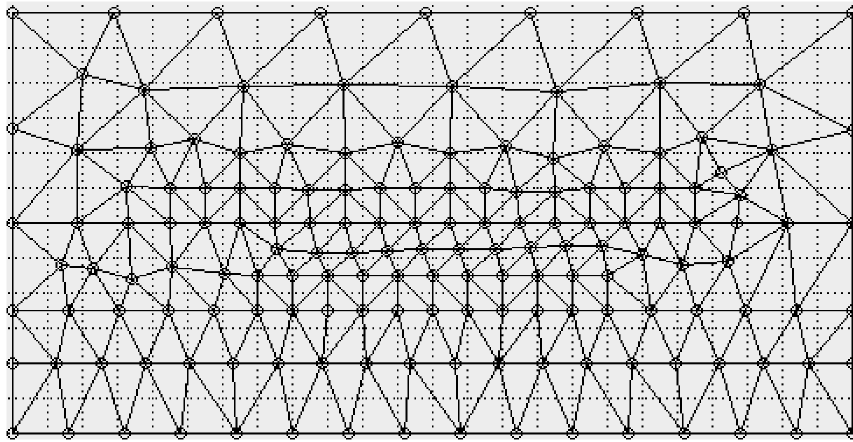


Figure 11: Gridded geometry

```
Calculated Capacitances:
------------------------
  1 C(1,2)=     C(boundary0,boundary1)= 2.866075e-10 F/m
  2 C(1,3)=     C(boundary0,boundary2)= 3.442174e-11 F/m
  3 C(1,4)=     C(boundary0,boundary3)= 1.867438e-12 F/m
  4 C(1,5)=     C(boundary0,boundary4)= 3.443857e-11 F/m
  5 C(2,3)=     C(boundary1,boundary2)= 6.312006e-11 F/m
  6 C(2,4)=     C(boundary1,boundary3)= 1.188536e-10 F/m
  7 C(2,5)=     C(boundary1,boundary4)= 6.302753e-11 F/m
  8 C(3,4)=     C(boundary2,boundary3)= 2.887580e-11 F/m
  9 C(3,5)=     C(boundary2,boundary4)= 2.874463e-12 F/m
 10 C(4,5)=     C(boundary3,boundary4)= 2.885160e-11 F/m
```

Table 2: Calculated capacitances

index 2 and the three lines in the upper layer have indices 3 to 5.) Four simulation runs with different conductor potentials were necessary for this calculation. The potential distribution of run four is presented in Figure 12 as an example.

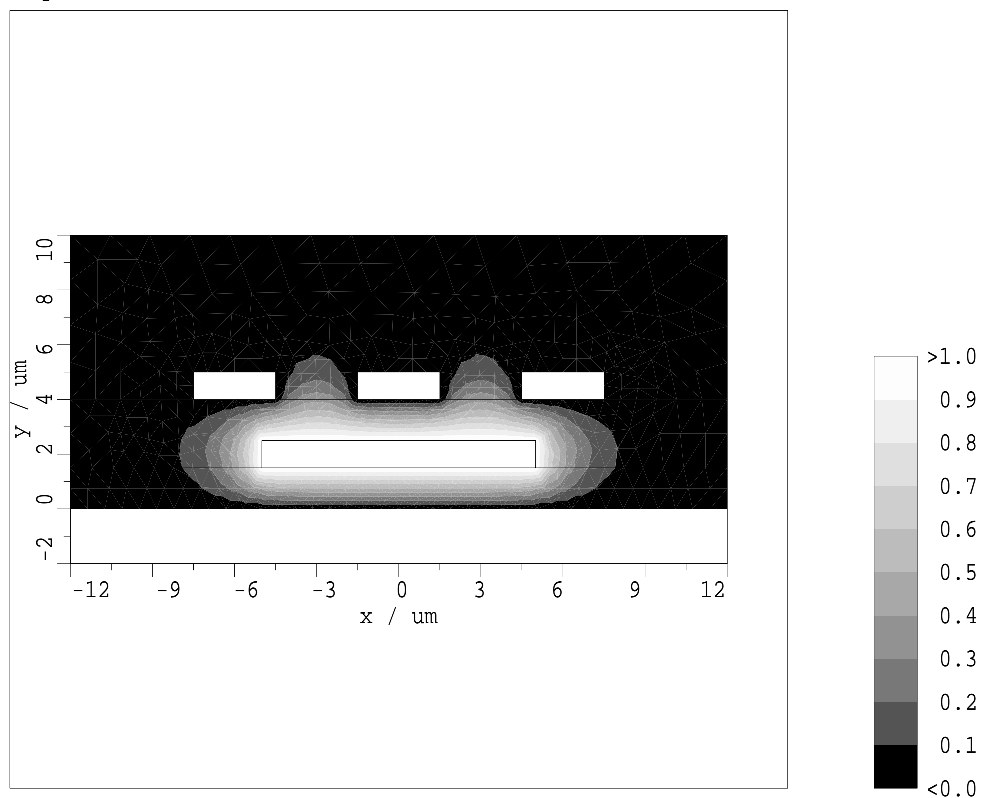


Figure 12: Potential distribution in the interconnect structure (second run)

4 Transient Simulation Using MINIMOS-NT

4.1 Features of the Transient Simulation

The following list shows the most important features of the transient simulation with MINIMOS-NT:

- Arbitrary number of controlled contacts.
- Transient voltages, currents and charges.
- Akima interpolation for contact values between specified instances.
- Predictor-corrector method for time step size estimation.

4.2 Time Dependent Contact Values

Arbitrary time dependent signals can be applied to a contact by specifying the signal value and information whether the signal is differentiable for each instance (time sample). The additional information is necessary for the correct interpolation of contact values between two instances.

4.3 Time Step Estimation

A prediction-correction method is used for the transient integration. The prediction uses either a linear or a quadratic time step size estimation. The linear estimation is used for the initial guess and after reaching instances containing a non differentiable contact value. Whenever two or more valid time steps have been made after a non differentiable instance the quadratic time step estimation is used.

The predictor used to estimate the size of the next time step is based on either a linear or a quadratic extrapolation of the potential-update norm (Figure 13).

Two different time step estimates are calculated using the L_2 norm ($\|\Delta\vec{\varphi}\|_2 = \sqrt{\sum_i \varphi_i^2}$) and the infinity norm ($\|\Delta\vec{\varphi}\|_\infty = \max_i \Delta\varphi_i$) of the potential-update, respectively. The smaller of the two

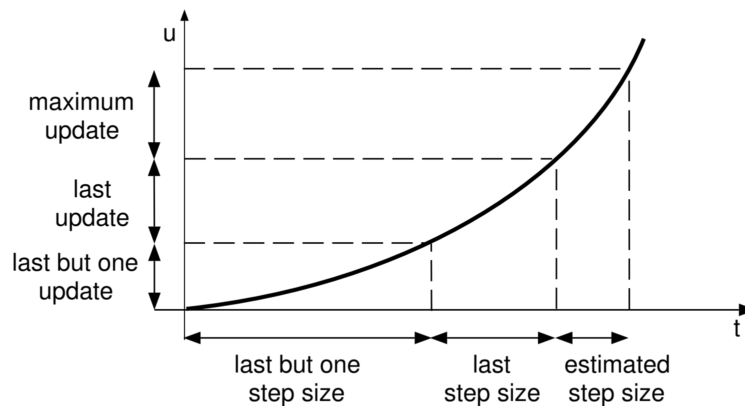


Figure 13: The quadratic time step estimation.

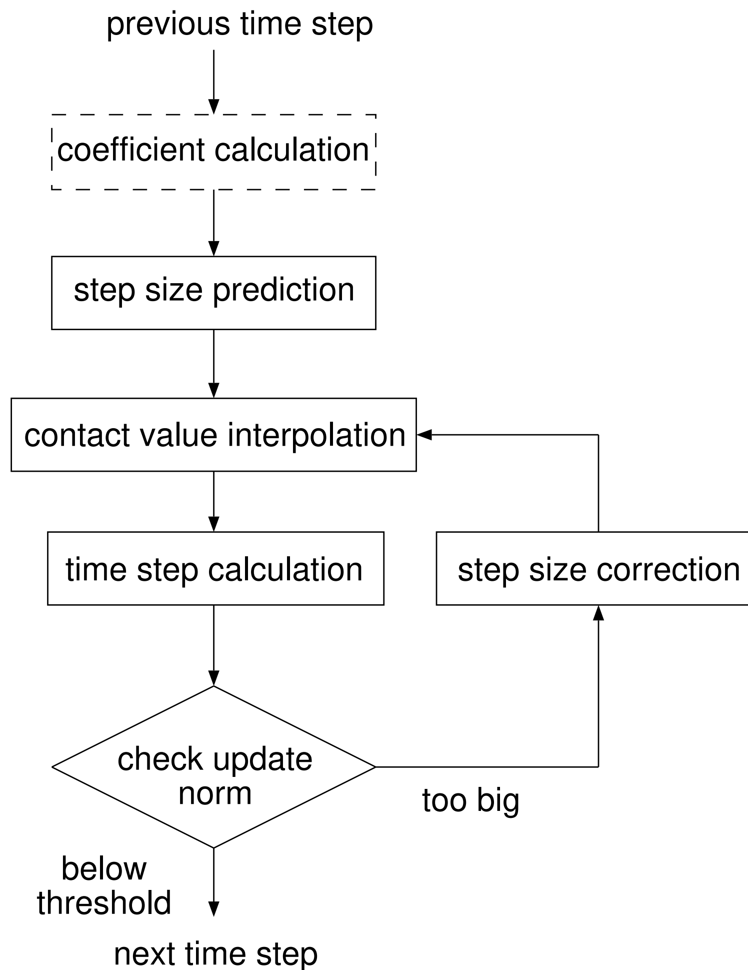


Figure 14: Control flow for a time step.

values is selected as the current time step estimate. Then the ratio of the estimated step size and the previous step size is restricted to a maximum, the sectio aurea constant (1.618...), to limit the step size variations. With this restriction a quasi-uniform time mesh is achieved, which gives a second-order local truncation error.

After calculation of each time step the potential-update is checked. For this purpose the L_2 norm and the infinity norm are calculated. If at least one of these norms exceeds a respective threshold, the step size is reduced by a linear or a quadratic interpolation, depending on which method has been used for prediction, and the calculation is repeated. Figure 14 shows the control flow for one time step.

In order to accurately follow the predefined contact signals it is necessary to calculate a time step at the instances (time samples) used to specify the contact signals. This requires a reduction of the estimated step size to exactly match the instance and causes an increase in the total number of required time steps (Figure 15). Therefore, the number of specified instances should be as small as possible to take full advantage of the quadratic time step estimation. To reduce the number of specified instances (time samples) which is necessary for a sufficiently accurate representation of nonlinear contact signals a higher-order interpolation has to be used.

When the estimated step size is smaller than the difference between the next instance and the actual time we check whether the double estimated step size is bigger than this difference (Fig-

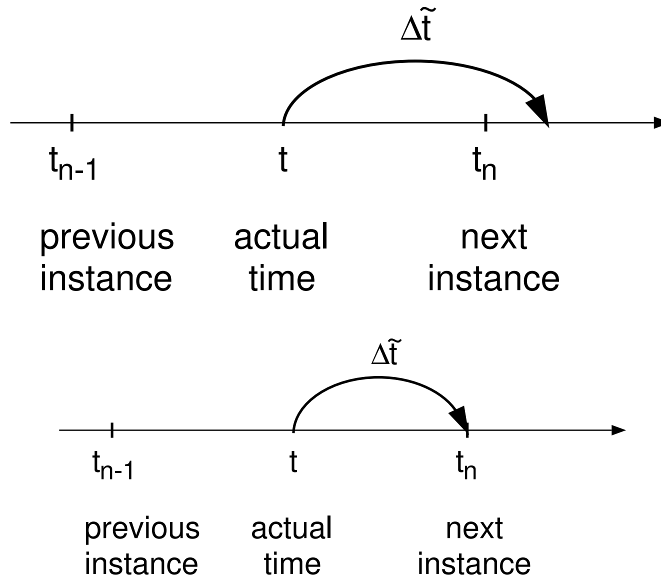


Figure 15: Reduction of the step size to exactly match an instance.

ure 16). If this is true then the time difference between the next instance and the actual time is divided into two steps in such a way that the ratio of the step sizes equals the sectio aurea constant. Without checking this condition, a big step could be followed by a sequence of much smaller steps because all subsequent step sizes could be at a maximum the sectio aurea constant times the previous step size.

4.4 Interpolation of Contact Values

For the interpolation of contact values between specified instances an Akima interpolation is used. This is a local third-order interpolation which guarantees a continuous first-order derivative. Because it is a local interpolation method there is no need to solve large equation systems in order to calculate the interpolation coefficients. At the beginning and at the end of a contact signal and at points where the contact signal is not differentiable the order of the interpolation function reduces to second order.

The requirement to reduce the number of specified instances to represent a nonlinear contact signal, which was mentioned in the previous section, is met very well by the Akima interpolation. Compared to higher-order interpolation functions it has the advantage that it does not tend to introduce oscillations.

4.5 Transient Simulation of a CCD

As an example a three-phase clock, n-channel charge coupled device (CCD) with fifteen gates has been simulated in two space dimensions [5]. Figure 17 shows the structure of the device. The source and drain contacts were held constant at 0V, the bulk contact at -1V. The voltages applied to the gates varied between -1V and $+5$V (Figure 18). The simulation over ten clock periods required approximately 1500 time steps. The adapted space grid consisted of approximately 10,000 points. A snapshot of the electron concentration in the channel region is shown

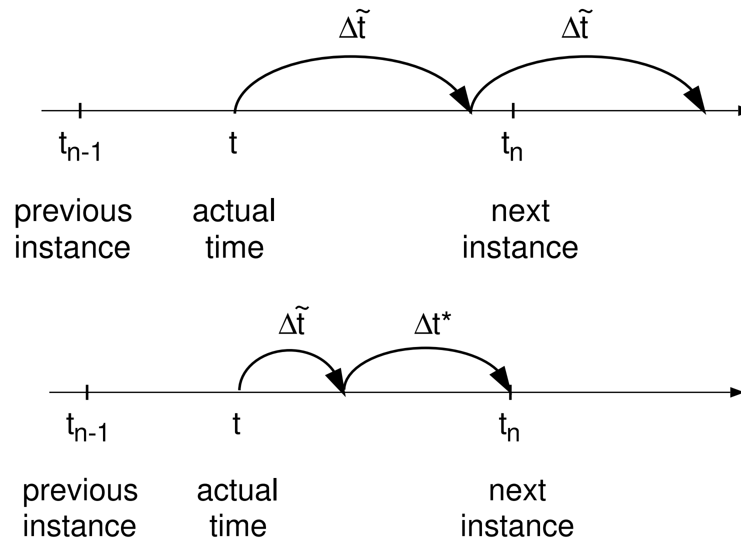


Figure 16: Reduction of the step size to exactly match an instance.

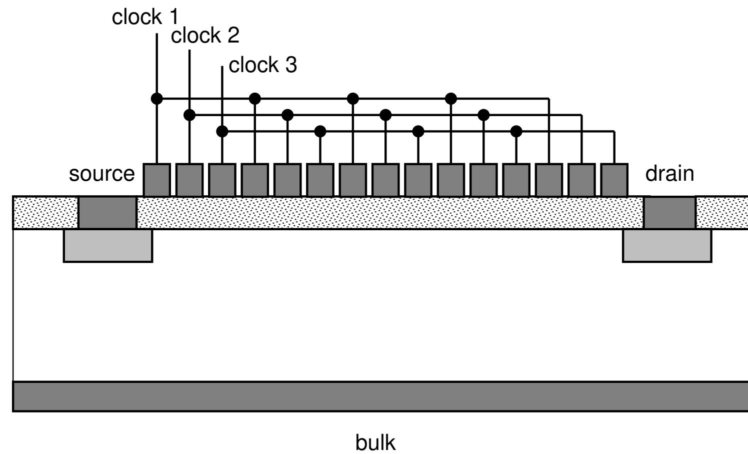


Figure 17: The structure of a 15-gate CCD.

in Figure 20. One can nicely see the alternation of accumulation to strong inversion under the gates. As a further result of the simulation Figure 5 shows the charge that has passed through the source, drain and bulk contact, respectively (charge flowing into the device is counted positive).

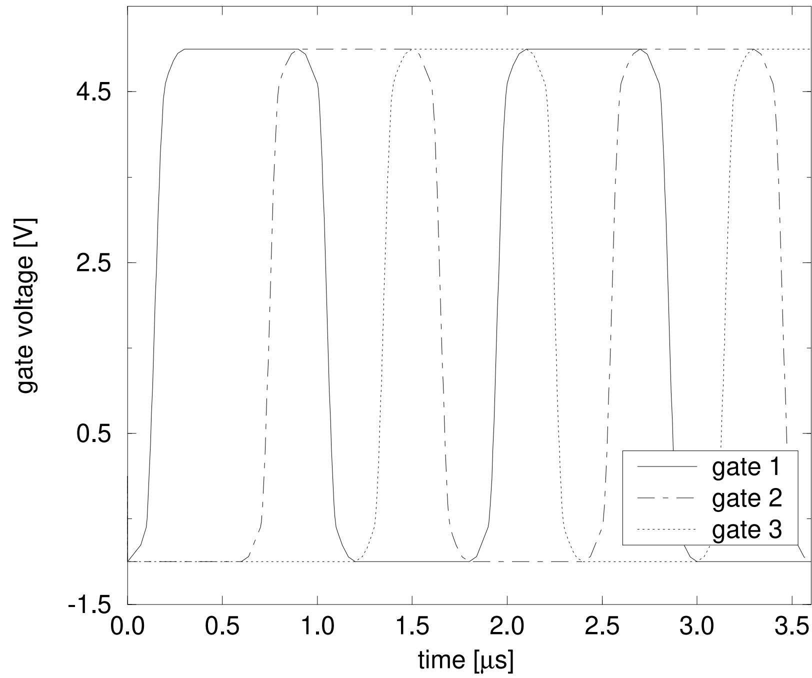


Figure 18: The clock voltages during the first two clock periods.

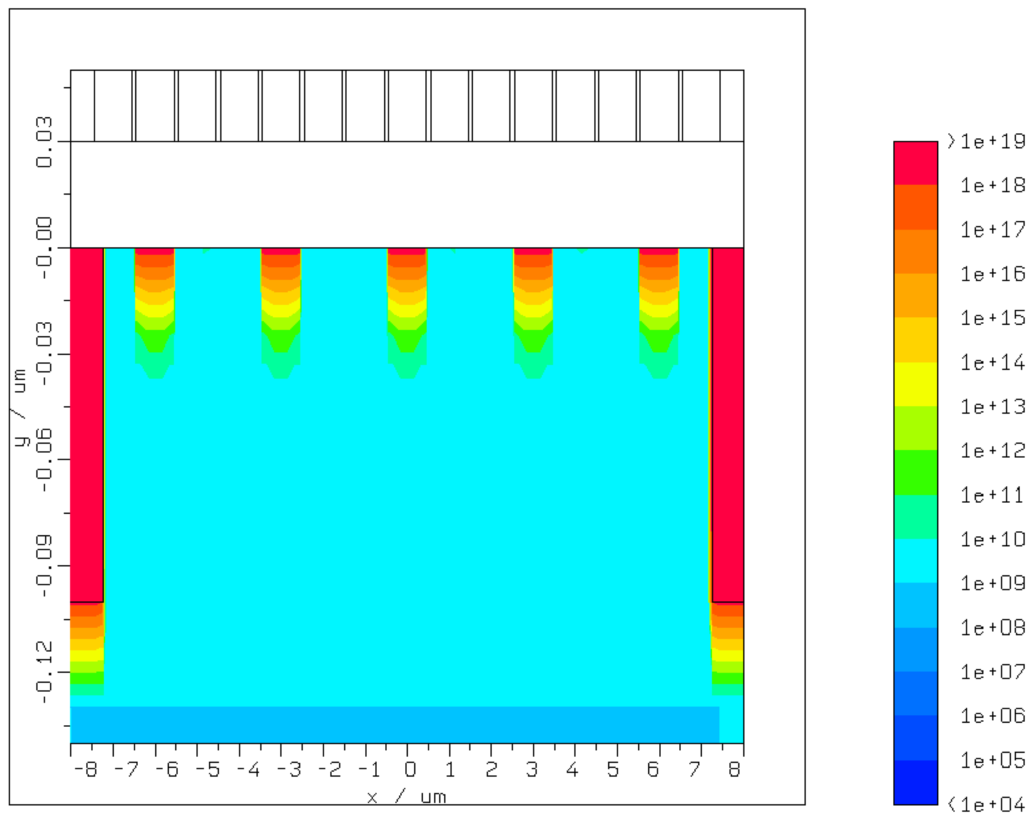


Figure 19: Electron concentration [cm^{-3}] in the channel region at $t = 17.47\mu\mathrm{s}$.

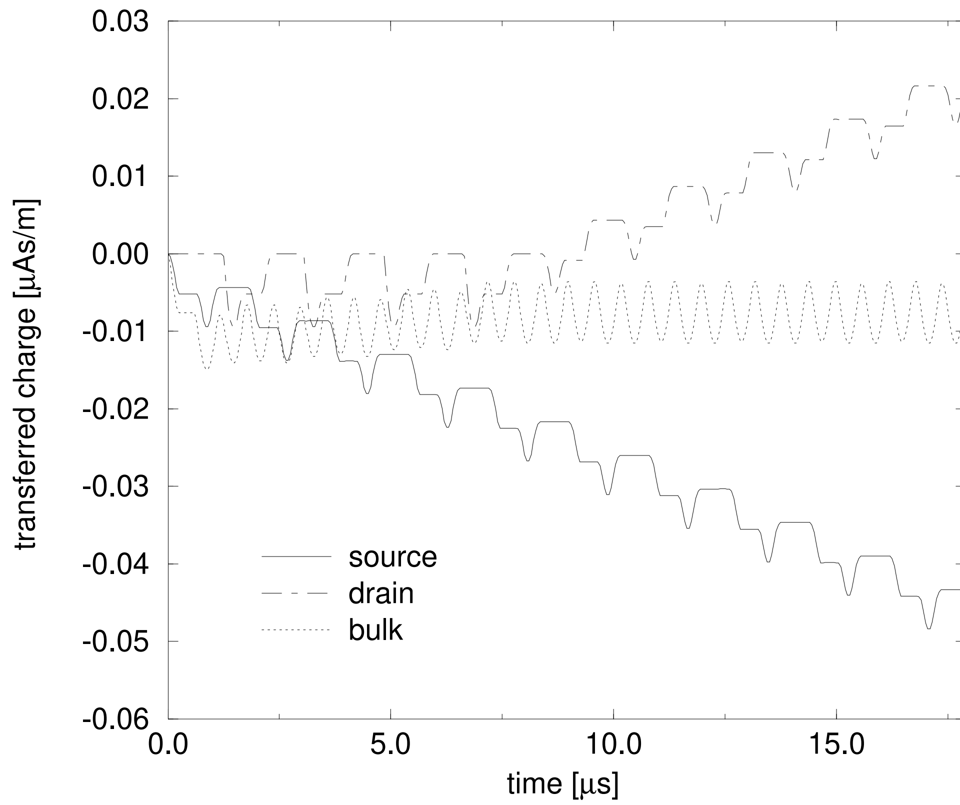


Figure 20: Charge transferred through the contacts of a 15-gate CCD during 10 clock periods.

5 Modeling of Thermal Effects with MINIMOS-NT

For the investigation of thermal effects with MINIMOS-NT the physical models have been improved to be valid in a temperature range from $77K$ to $500K$ [6, 7].

There are two methods to simulate thermal effects. The simplest one assumes a uniform lattice temperature in the device. The temperature dependence of the basic physical quantities such as band edge energies, densities of states, effective masses of carriers etc. is properly accounted for.

The more sophisticated method includes self-heating effects. The basic equations of both the drift-diffusion and the hydrodynamic model have been extended to deal with the spatially varying lattice temperature. In addition, the heat flow equation is solved for the lattice temperature as unknown. Each of the temperature dependent physical parameters has to be recalculated after each Newton-iteration.

In the following sections we describe the heat flow equation and the boundary conditions as well as the global energy balance of a device. In the final section results of a thermal simulation of a MOSFET are presented.

5.1 Differential Equations of the Lattice Heat Flow

For the heat flow equation in the lattice two different heat generation terms have to be considered, depending on the underlying carrier transport model.

$$-\operatorname{div}(\kappa \text{ grad } T_L) = \vec{E} \cdot (\vec{J_n} + \vec{J_p}) \tag{5}$$

$$-\operatorname{div}(\kappa \text{ grad } T_L) = \frac{3}{2} k_B \left(n \cdot \frac{T_n - T_L}{\tau_{\epsilon,n}} + p \cdot \frac{T_p - T_L}{\tau_{\epsilon,p}} \right) \tag{6}$$

The first equation applies to the drift-diffusion and the second one to the hydrodynamic case. Thermal conductivity is denoted by κ, and $\tau_{\epsilon,n}$ $\tau_{\epsilon,p}$ are the energy relaxation times for electrons and holes, respectively. T_n and T_p represent the temperatures of the electron and hole systems, respectively, and T_L is the lattice temperature.

5.2 Boundary Conditions

By default all electrical contacts are treated as thermal contacts as well. To realize a boundary as a thermal contact only, the user has to specify an electrical insulator with the desired thermal properties at present. Currently, three types of boundary conditions are available for the heat flow equation.

Default condition: If no temperature is specified on a contact a homogeneous NEUMANN boundary condition is assumed. This means that no thermal heat is exchanged with the ambient over this contact.

Constant contact temperature: A constant contact temperature, which has to be specified by the user in the input PIF-File, imposes a DIRICHLET boundary condition on the heat flow equation.

Thermal resistance: By means of a thermal resistance R_{th} the normal component of the heat flux density can be specified representing an CAUCHY boundary condition:

$$\vec{S} \cdot \vec{n} = \frac{T_L - T_R}{R_{th}} \tag{7}$$

$\vec{n}$ is a unity vector normal to the boundary, and T_R is the temperature of the surrounding heat sink. The parameters T_R and R_{th} need to be defined in the input PIF-File.

5.3 Thermal Energy Balance of a Device

The device can be considered as an open system. The energy flowing into the system is the sum of all contact voltages times the contact currents. In the stationary case this energy is equal to the heat flux over all thermal contacts.

As an example a forward biased diode has been simulated. The terminal quantity information written to the simulator log file is shown below to demonstrate that the global energy balance is fulfilled.

```
                  Terminal quantities information:
                  ================================
Line Step
TH 1    0      Voltage/S2[V]     Current/S2[A]      Charge/S2[C]   Heat Flux/S2[W]
TH 2    0      Voltage/S3[V]     Current/S3[A]      Charge/S3[C]   Heat Flux/S3[W]
TH 3    0     Current Sum[A]  Heat Flux Sum[W]      Sum (U.I)[W]
TI 1    0  0.0000000000e+00  6.5163673335e-03  2.9409239096e-16  8.6165244160e-03
TI 2    0 -2.0000000000e+00 -6.5163673335e-03 -3.6234420766e-18  4.4162448727e-03
TI 3    0  0.0000000000e+00  1.3032769289e-02  1.3032734667e-02
```

The quantities labeled 'Heat Flux Sum' (1.303...) and 'Sum (U.I)' (1.303...) are found to agree very well.

For the correct energy balance it is necessary to take into account the contact potential between the ideal conductor and the semiconductor. This leads to a step in the electrostatic potential which causes a change of the carrier energy when crossing the boundary.

5.4 Input Data for a Thermal Simulation

The self-heating mode is invoked by the command line option '-LT'. The following example of a PIF-Input file illustrates the specification of thermal contact properties.

```
(attribute attribute_7
  (attributeType "SegmentDescription")
  (nameList
    (ref segments_1
      (valueList 4)))
  (attribute attribute_8
    (attributeType "MaterialType")
    (valueType asciiString)
    (valueList "Al"))
  (attribute AnodeContactTemperature
```

```
      (attributeType "ContactTemperature")
      (valueType real)
      (units "K")
      (valueList +300.00000000000000      ))
    (attribute AnodeContactResistance
      (attributeType "ThermalSpreadingResistance")
      (valueType real)
      (units "K m^2/W")
      (valueList +5.0000000000000002e-13 )))
```

At the beginning of the simulation the lattice temperature is set to a constant value in each segment. This value is determined by averaging the specified contact temperatures. This temperature is then used to calculate temperature dependent values such as band edge energies, electron and hole masses and the effective density of states.

5.5 Example

Simulation of self-heating in a one-micron gate length n-channel MOSFET has been carried out. Bias voltages are $V_{GS} = V_{DS} = 6V$. The thermal resistances for each contact are $R_{th} = 1.5 \cdot 10^{-4} Kcm^2/W$. In Figure 21 the temperature distribution of the transistor is shown. The major part of the dissipated energy is transferred to the bulk contact. It should be noticed that the contacts are non-isothermal surfaces.

Figure 21: Lattice temperature in Kelvin of a MOS-Transistor with $U_{GS} = U_{DS} = 5$V.

References

[1] G. Schumicki and P. Seegebrecht. *Prozeßtechnologie*. Springer, 1991.

[2] G. Rieger, S. Halama, and S. Selberherr. A Programmable Tool for Interactive Wafer-State Level Data Processing. In Ryssel and Pichler [8], pp 58–61.

[3] G. Rieger and S. Selberherr. The PIF Editor – a Data Processor for the VISTA TCAD Framework. In *Proc. HPC'ASIA (CD)*, Taipei, 1995.

[4] R. Bauer. *Numerische Berechnung von Kapazitäten in dreidimensionalen Verdrahtungsstrukturen*. Dissertation, Technische Universität Wien, 1994.

[5] M. Rottinger, T. Simlinger, and S. Selberherr. Two-Dimensional Transient Simulation of Charge-Coupled Devices Using MINIMOS NT. In Ryssel and Pichler [8], pp 440–443.

[6] H. Brand. *Thermoelektrizität und Hydrodynamik*. Dissertation, Technische Universität Wien, 1994.

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

[8] H. Ryssel and P. Pichler, editors. *Simulation of Semiconductor Devices and Processes*, volume 6. Springer, 1995.