SILVACO International

TCAD
Tutorial and Examples
Volume III

SILVACO International
4701 Patrick Henry Drive, Bldg. 1
Santa Clara, CA 94054
Telephone: (408) 567-1000
FAX: (408) 496-6080
E-Mail: support@silvaco.com
Internet: http://www.SILVACO.com

Edition Two
January 1999

TCAD
Tutorial and Examples Manual
Volume III

SILVACO International
4701 Patrick Henry Drive, Bldg. 1
Santa Clara, CA 95054

Phone: (408) 567-1000
FAX: (408) 496-6080
E-Mail: support@silvaco.com
Internet: http://www.SILVACO.com

Notice

The information contained in this document is subject to change without notice.

SILVACO International MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE.

SILVACO International Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

SIMULATION STANDARD, TCADDRIVENCAD, VIRTUAL WAFER FAB, ANALOG ALLIANCE, LEGACY, ATHENA, ATLAS, FAST ATLAS, ODIN, VYPER, CRUSADE, RESILIENCE, DISCOVERY, CELEBRITY, PRODUCTION TOOLS, AUTOMATION TOOLS, INTERACTIVE TOOLS, TONYPLOT, DECKBUILD, DEVEDIT, INTERPRETER, ATHENA INTERPRETER, ATLAS INTERPRETER, CIRCUIT OPTIMIZER, MASKVIEWS, PSTATS, SSUPREM3, SSUPREM4, ELITE, OPTOLITH, FLASH, SILICIDES, SPDB, CMP, MC DEPOSIT, MC IMPLANT, PROCESS ADAPTIVE MESHING, S-PISCES, BLAZE, DEVICE 3D, INTERCONNECT3D, BLAZE3D, GIGA3D, MIZEDMODE3D, TFT, LUMINOUS, GIGA, MIZEDMODE, ESD, LASER, FASTBLAZE, FASTMIXEDMODE, FASTGIGA, FASTNOISE, MOCASIM, UTMOST, UTOMST II, UTMOST III, UTMOST IV, PROMOST, SPAYN, SMARTSPICE, MIXSIM, TWISTER, FASTSPICE, SMARTLIB, SDDL, EXACT, CLEVER, STELLAR, HIPEX, LISA, SCHOLAR, SIREN, ESCORT, STARLET, EXPERT, SAVAGE, SCOUT, GUARDIAN, and ENVOY are trademarks of SILVACO INTERNATIONAL.

All other trademarks mentioned in this manual are the property of their respective owners.

Reader Comment Sheet

We welcome your evaluation of this manual. Your comments and suggestions help us to improve our publications. If you have any responses to the questions below, please let us know. Write your observations down and send complaints, bug reports, suggestions or comments to the e-mail address listed below.

- Is this manual technically accurate?
- Are the concepts and wording easy to understand?
- Is the size of this manual convenient for you?
- Is the manual's arrangement convenient for you?
- Do you consider this manual to be easily readable?

Please add any additional relevant comments.

Please FAX your comments to

SILVACO International
Attention Technical Publications
1701 Patrick Henry Drive, Building 1
Santa Clara, CA 95054
at (408) 496-6080.

or

Send e-mail to us at: *support@silvaco.com*

or

Browse our Web Page at *http://www.SILVACO.com*

Intended Audience

The information in this manual is based on the following assumptions:

- The reader is familiar with the basic terminology of semiconductor processing and semiconductor device operation,

  and

- The reader understands the basic operations of the computer hardware and operation systems being used.

Introduction

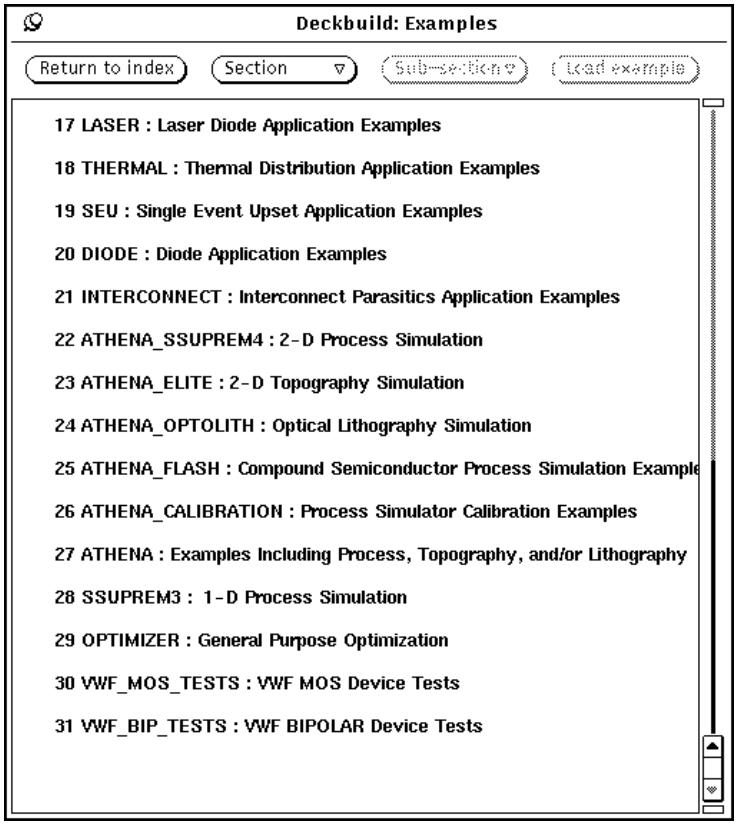
This manual is intended as an additional guide to the use of Silvaco's process and device simulators. It contains descriptions of all the standard examples that demonstrate the use of SSUPREM3, ATHENA, ATLAS, and the VWF INTERACTIVE TOOLS manuals. Users should consult the relevant "User's Manual" for a full description of the models and syntax of each program.

Included on your distribution media are more than five hundred (500) Standard Examples that demonstrate the way that the simulators are used to model many different technologies. The examples are instructional and it is strongly recommended that new users apply these examples as a starting point for creating their own simulations. One of the first things you should learn is how to access, load, and run these examples.

Accessing the Examples

The examples are accessed from the menu system in DECKBUILD. To select and load an example:

1. Start DECKBUILD as described in the VWF INTERACTIVE TOOLS MANUAL.
2. Pull down the **MainControl** menu using the right hand mouse button. There are options on this menu for **MainControl**, **Optimizer**, **Examples**, **Help**, etc.
3. Select **Examples**. An index will appear in a **DeckBuild: Examples** window (see below). The examples are divided by technology or technology group. The most common technologies are clear (e.g., MOS, BJT) while others are grouped with similar devices (e.g., IGBT and LDMOS are under POWER, and solar cell and photodiode are under OPTOELECTRONICS).



The Examples Index in DeckBuild

4. Choose the technology you are interested in by double-clicking the left mouse button over that item in the examples index.

5. A list of examples for that technology will appear. These examples typically illustrate different devices, applications, or types of simulation.

6. Choose a particular example by double-clicking the left mouse button over that item in the list.

7. A text description of the example will appear in the window. This online text is the same as in this manual. It describes the important physyical mechanisms in the simulation, as well as giving details of the simulator syntax used. You should read this information before proceeding.

8. Press the **Load Example** button. The **Input Command** file for the example will be copied into your current working directory, together with any associated files. A copy of the command file will be loaded into DECKBUILD. (Note that the **Load Example** button remains faded until **Step 6** is performed correctly.

9. To run the example, press the **Run** button in the middle frame of the DECKBUILD application window.

10. Alternatively, most examples are supplied with results that can be copied into the current working directory, along with the input file. To view the results, select (highlight) the name of the **Results File** and select the DECKBUILD menu option, **Tools-Plot**. Details on the use of TONYPLOT can be found in the VWF INTERACTIVE TOOLS manual.

Examples Index:

This page is intentionally left blank

11.1. MESFET: MESFET Application Examples

11.1.1 mesfetex01.in: Ion Implanted GaAs MESFET Fabrication and Vt Test

Requires: FLASH/DevEdit/BLAZE

This example demonstrates fabrication and electrical analysis of a MESFET structure using the FLASH module of ATHENA and the BLAZE capability of ATLAS. The example uses **DevEdit** at various points in the process to optimize the grid. The example shows:

- MESFET fabrication using FLASH
- re-meshing of structure in DevEdit
- setting of device parameters for electrical simulation
- electrical simulation of an Id/Vgs curve

This example starts by interfacing ATHENA and DEVEDIT to provide a GaAs MESFET structure using silicon and beryllium implants. Details of using FLASH can be found in the ATHENA_FLASH examples section.

It is necessary to set contact workfunctions for every Schottky contact in each ATLAS input file. The command, `contact name=gate work=4.87`, is used to set the gate workfunction. Source and drain contacts are assumed to be ohmic. Users should note that this contact statement is necessary even though the material Titanium was used in the ATHENA simulation. ATLAS does not recognize different metals and the properties of different metal regions are not transferred from ATHENA to ATLAS. ATLAS calculates the barrier height of a Schottky barrier in the following manner:

```
barrier height = (metal work function - semiconductor electron affinity).
```

Care should be taken to ensure the appropriate barrier height is chosen.

Low lifetimes typical of GaAs are set in the material statement. The models statement is used to specify appropriate models within the simulation. The fldmob parameter turns on the electric field dependent mobility. Conmob specifies the concentration dependent mobility. These values are taken from a look-up table and exist only for room temperature.

The solution sequence for MESFET threshold voltage is first to obtain the initial solution at zero bias on all contacts. Then the drain is set to 0.1V. A log file is opened by the **log** statement to store all terminal characteristics. The final solve statement ramps the gate from zero to -3V. Note that ATLAS requires the zero bias solution to be solved first in all cases. Thus the gate voltage should be swept from zero to the value required. If a range of both negative and positive gate voltages is required two sweeps can be done by saving and loading the zero bias solution.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

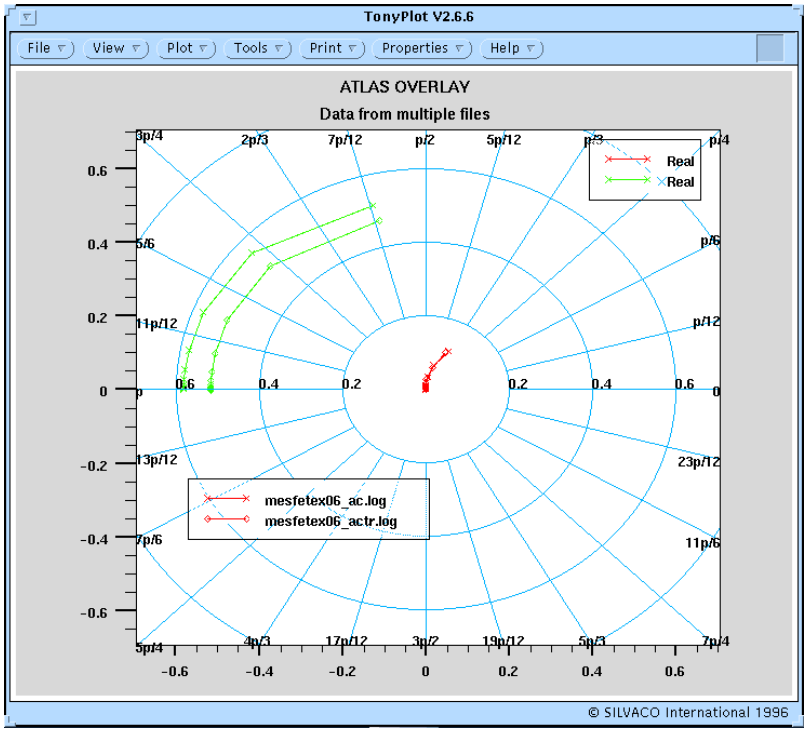


Figure 11.1: Structure and Doping profile of GaAs MESFET simulated using FLASH

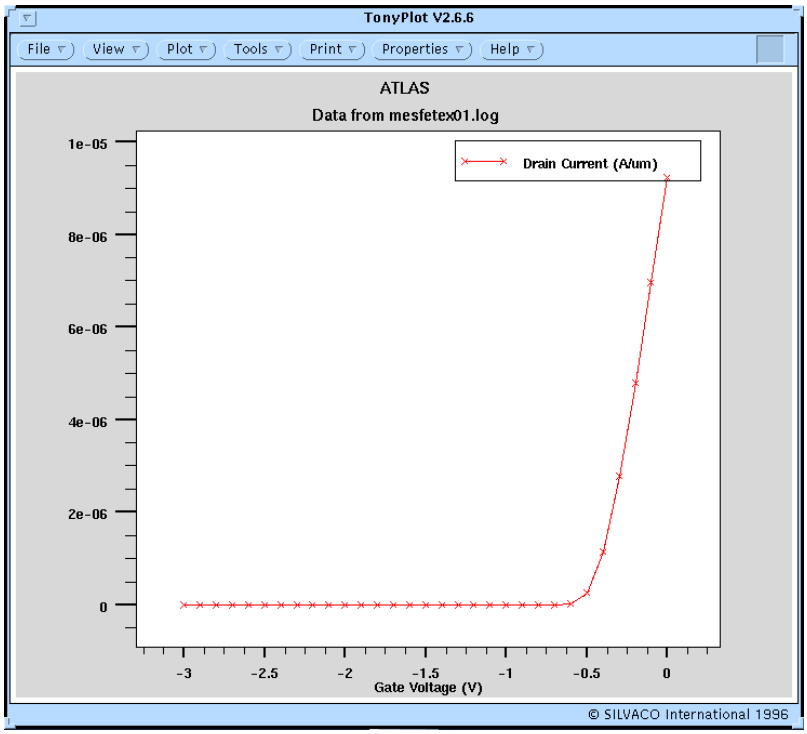


Figure 11.2: Id/Vgs curve for the MESFET showing the threshold voltage

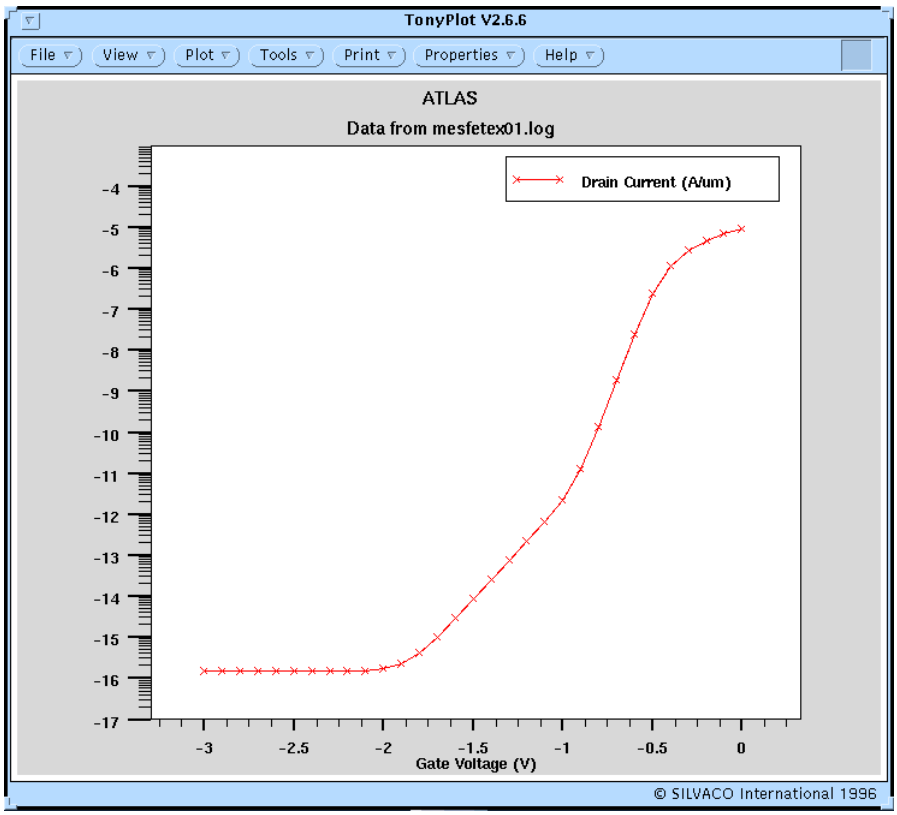


Figure 11.3: Sub-threshold behavior of the MESFET

Input File mesfet/mesfetex01.in:

```
1   go athena
2
3   # GaAs MESFET fabrication and analysis using DevEdit
4
5   line x loc=-1.5 spac=0.2
6   line x loc=-.7  spac=0.1
7   line x loc=-.5  spac=0.05
8   line x loc=0.0  spac=0.1
9   line x loc=0.5  spac=0.05
10  line x loc=0.7  spac=0.1
11  line x loc=1.5  spac=0.2
12  #
13  line y loc=0.00 spac=0.02
14  line y loc=2.00 spac=0.5
15  #
16  init gaas c.beryllium=1.0e13 orient=100 space.mult=1
17
18  implant beryllium energy=100 dose=2e11
19  implant silicon energy=100 dose=1e12
```

```
20
21 diffus time=10 temp=850
22
23 # deposit and pattern gate metal
24 deposit titanium thick=.3 divisions=10
25 etch titanium right p1.x=0.5
26 etch titanium left  p1.x=-0.5
27
28 deposit oxide thick=0.35 divisions=8
29
30 etch oxide thick=.4
31
32 # regrid before implant
33 go DevEdit
34
35 base.mesh height=0.08 width=0.08
36 bound.cond apply=false max.ratio=300
37 bound.cond when=automatic max.slope=28 rnd.unit=0.001 line.straighten-
     ing=2 \
38        align.points
39 constr.mesh max.angle=90 max.ratio=300 max.height=1 max.width=1 \
40        min.height=0.0001 min.width=0.0001
41 constr.mesh type=Semiconductor default
42 constr.mesh type=Insulator default
43 constr.mesh type=Metal default
44
45
46 imp.refine min.spacing=0.02
47 imp.refine imp="net doping" sensitivity=1
48 mesh
49
50 go athena
51
52 # perform source/drain implant
53 implant silicon energy=50 dose=1e13
54
55 # regrid to reduce grid in un-implanted areas
56 go DevEdit
57
58 base.mesh height=0.4 width=0.4
59 bound.cond apply=false max.ratio=300
60 bound.cond when=automatic max.slope=28 rnd.unit=0.001 line.straighten-
     ing=2 \
61        align.points
```

```
62  constr.mesh max.angle=90 max.ratio=300 max.height=1 max.width=1 \
63          min.height=0.0001 min.width=0.0001
64  constr.mesh type=Semiconductor default
65  constr.mesh type=Insulator default max.angle=180
66  constr.mesh type=Metal default
67
68  imp.refine min.spacing=0.03
69  imp.refine imp="net doping" sensitivity=.5
70  mesh
71
72
73  go athena
74
75  diff time=10 temp=850
76
77  # deposit ohmic metal
78  deposit aluminum thick=.2 divisions=4
79
80  etch aluminum        start x=-1 y=10
81  etch cont                  x=-1 y=-10
82  etch cont                  x=1 y=-10
83  etch done                  x=1 y=10
84  #
85  electrode name=source x=-1.4
86  electrode name=drain x=1.4
87  electrode name=gate x=0.0
88
89  # regrid to resolve new junction position
90  go DevEdit
91
92  base.mesh height=0.1 width=0.1
93  bound.cond apply=false max.ratio=300
94  bound.cond when=automatic max.slope=28 rnd.unit=0.001 line.straighten-
      ing=2 \
95          align.points
96  constr.mesh max.angle=90 max.ratio=300 max.height=1 max.width=1 \
97          min.height=0.0001 min.width=0.0001
98  constr.mesh type=Semiconductor default
99  constr.mesh type=Insulator default max.angle=180
100 constr.mesh type=Metal default
101
102 imp.refine min.spacing=0.03
103 imp.refine imp="net doping" sensitivity=1
```

```
104 mesh
105
106 structure outfile=mesfetex01_0.str
107 tonyplot  mesfetex01_0.str  -set mesfetex01_0.set
108
109 go atlas
110
111
112 # set work function for gate
113 contact name=gate work=4.87
114
115 # specify lifetimes in GaAs and models
116 material material=GaAS taun0=1.e-8 taup0=1.e-8
117 models   conmob fldmob srh optr print
118
119 # Begin solution
120 method newton trap
121 solve vdrain=0.1
122
123 # Ramp gate and log results
124 log    outf=mesfetex01.log master
125 solve vgate=0.0 vstep=-0.1 vfinal=-3 name=gate
126
127 extract init infile="mesfetex01.log"
128 extract name="vt" (xintercept(maxslope(curve((v."gate"),(i."drain")))))
129
130 save outfile=mesfetex01_1.str
131
132 tonyplot  mesfetex01.log
133
134 quit
135
136
137
138
139
```

11.1.2 mesfetex02.in: Epitaxial GaAs MESFET S-Parameters

Requires: BLAZE

This example demonstrates the calculation of s-parameters in a simple MESFET. It shows:

- MESFET structure definition using ATLAS syntax
- setting of GaAs mobility models and gate workfunction

- Id/Vds characteristics with Vgs=0
- AC analysis at a single DC bias at different frequencies
- conversion of capacitance and conductance data from ATLAS into s-parameters
- Smith Charts of MESFET s-parameters using TonyPlot

The geometry and doping of the MESFET device used in this example is described using the ATLAS structural syntax. Initially the mesh is specified in x and y, using the following command format `x.mesh loc=aaa spac=bbb` specifying that there is a mesh spacing of bbb and x location aaa. `Region` statements are used to specify two GaAs regions which will later be doped differently. `Electrode` statements specifies the names and positions of electrodes. The workfunction of the gate contact is set using `contact`. Each GaAs region is uniformly doped. An n-type active layer sits on top of a lowly doped p-type substrate. GaAs mobility models for concentration and field dependence are set in the `models` statement.

The numerical methods used at the initial stage of the simulation are conservative. The statement, `method gummel newton` specifies that the decoupled (gummel) method is used at the start of the simulation for each bias point. This switches to the coupled (Newton) method if convergence is not obtained. This is the most robust method for the initial bias steps and for complex devices. It does consume more CPU time and can lead to problems at high current levels. If these problems occur the statement `method newton` can be used.

The DC simulation proceeds by ramping the drain voltage in the `solve` statement until Vds=3.0V. After this a log file is opened and AC analysis begins. The full sweep of frequency is done on the line:

```
solve ac.analysis direct frequency=1.e9 fstep=2.e9 nfsteps=20
```

The 'ac' parameter switches on the AC analysis. Direct specifies a robust AC solver for high frequencies. Activating the 'direct' solver will cause more memory to be allocated than was used during the DC simulation. If your system does not have enough virtual memory to run the AC application the simulation will stop here with an error message. The exact amount of memory required depends on the number of mesh points. The `frequency` value is the initial frequency of the AC signal, fstep is the step value added to this frequency and nfsteps is the number of steps.

The AC analysis in ATLAS calculates the real and imaginary current components from a small AC signal on top of the existing DC solution. From these currents ATLAS calculates the conductance and capacitance between each pair of electrodes. By plotting the log file, users can see graphs of drain-gate conductance (C drain>gate) or total gate capacitance (-1*C gate>gate).

The s-parameters (or `Z`, `Y` and `ABCD` parameters) are calculated by setting the **s.param** (or y.param etc) on the log statement preceeding the ac analysis. The most important parameters are the ***port** parameters. s-parameter analysis assumes the device is a two port device with four terminals. The inport and outport are used to assign the respective electrodes. Users should ensure the correct definition of all ports. In this example, the input port is set to the gate and output port to the drain.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

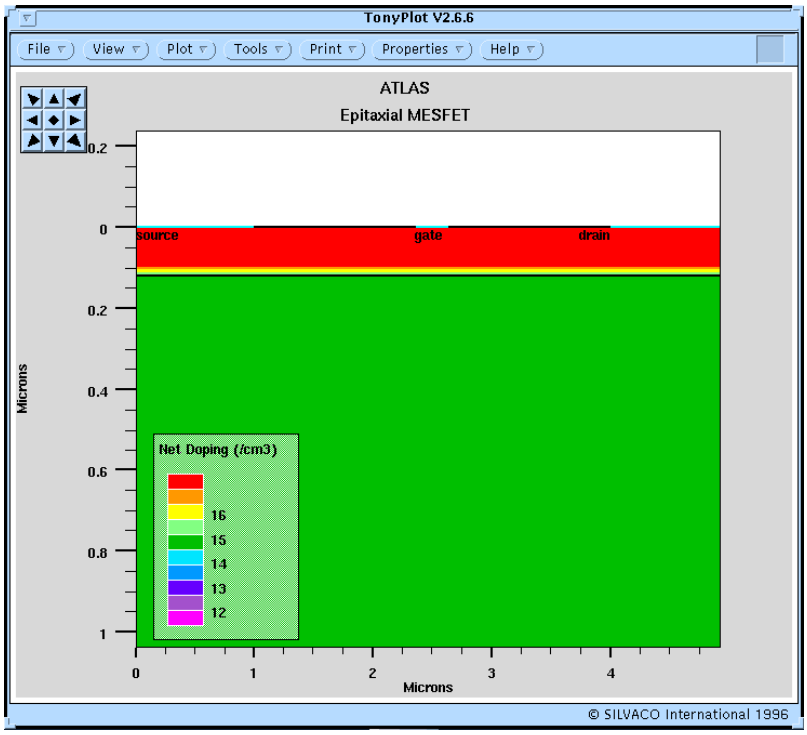


Figure 11.4: Doping profile of an Epitaxial MESFET defined using ATLAS syntax

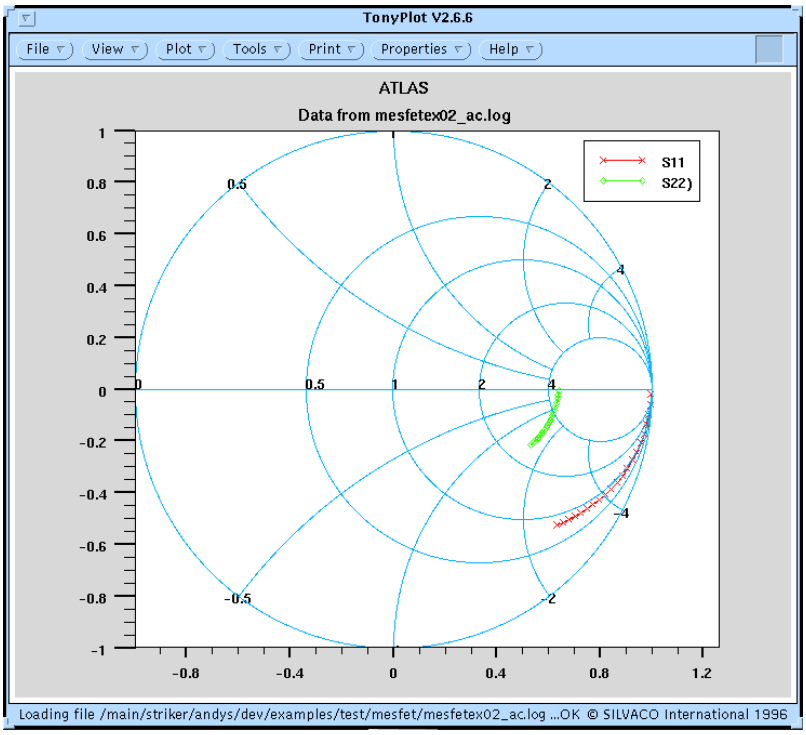


Figure 11.5: S11 and S22 from the MESFET displayed on a Smith Chart

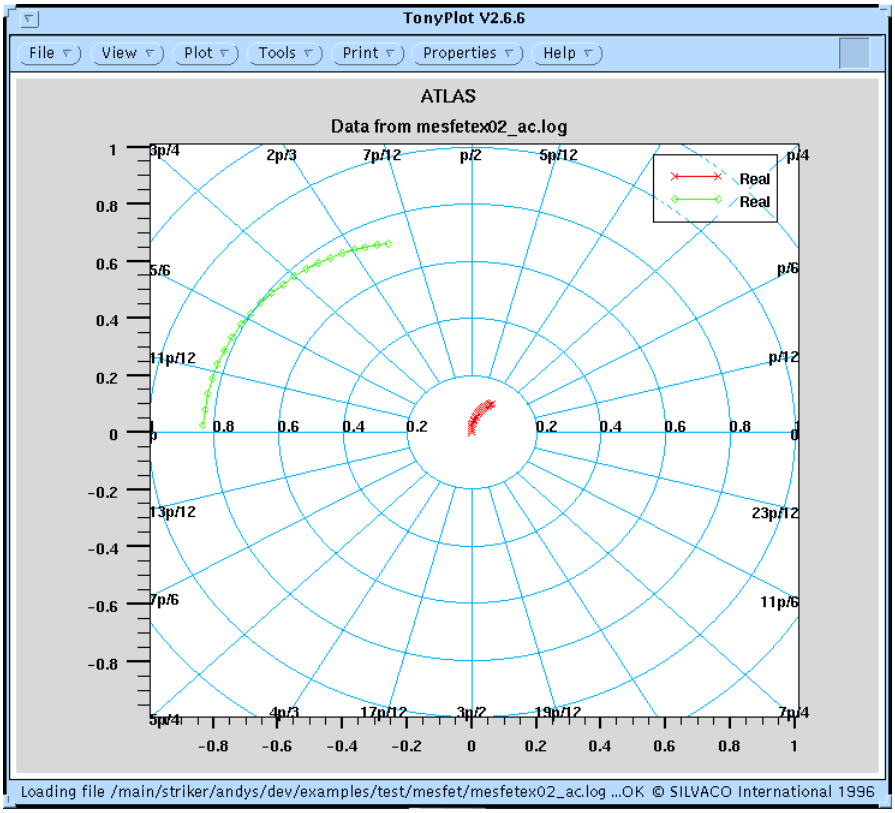


Figure 11.6: S12 (x) and S21 (o) displayed on a Polar plot

Input File mesfet/mesfetex02.in:

```
go atlas

Title  MBE Epitaxial GaAs MESFET - S parameters calculation

# Define the mesh

mesh  space.mult=1.0
#
x.mesh loc=0.00 spac=0.3
x.mesh loc=2.3 spac=0.02
x.mesh loc=2.7 spac=0.02
x.mesh loc=5 spac=0.3
#
y.mesh loc=0.00 spac=0.01
y.mesh loc=0.04 spac=0.03
y.mesh loc=0.12 spac=0.02
y.mesh loc=6 spac=1.0

# Region specification: Defined as two regions for different properties
```

```
20
21 region     num=1 GaAs x.min=0 x.max=5 y.min=0 y.max=0.12
22 region     num=2 GaAs x.min=0 x.max=5 y.min=0.12 y.max=6.12
23
24 # Electrode specification
25
26 elec       num=1  name=source  x.min=0.0 y.min=0.0 x.max=1.0 y.max=0.0
27 elec       num=2  name=drain   x.min=4.0 y.min=0.0 x.max=5.0 y.max=0.0
28 elec       num=3  name=gate    x.min=2.35 length=0.3
29
30 # Doping specification
31
32 doping region=1 uniform conc=1.0e17 n.type
33 doping region=2 uniform conc=1.0e15 p.type
34
35 # Set models, material and contact parameters
36
37 contact num=3 work=4.77
38 models region=1 print conmob fldmob srh optr
39 models region=2 srh optr
40 material region=2
41
42 # Solution - use gummel newton for initial then switch to full newton
43
44 method gummel newton
45 solve  vgate=0
46 save outfile=mesfetex02_1.str
47 tonyplot  mesfetex02_1.str -set mesfetex02_0.set
48 solve vdrain=0.025 vstep=0.025 vfinal=0.1 name=drain
49
50 method  newton
51 solve vdrain=0.2 vstep=0.1 vfinal=0.6 name=drain
52 solve vdrain=0.8 vstep=0.2 vfinal=3 name=drain
53
54 # Small signal ac analysis with s-parameter calculation
55 log outf=mesfetex02_ac.log s.param inport=gate outport=drain width=100
56 solve ac.analysis direct frequency=1.e9 fstep=2.e9 nfsteps=20
57
58 tonyplot  mesfetex02_s.log -set mesfetex02_1.set
59 tonyplot  mesfetex02_s.log -set mesfetex02_2.set
60
61 quit
```

11.1.3 mesfetex03.in: Energy Balance and Drift Diffusion Comparison

Requires: BLAZE

This example demonstrates electrical analysis of a 0.2um MESFET structure with Energy Balance (EB) and Drift-Diffusion (DD) Models. It shows:

- structure definition using ATLAS syntax
- specification of energy balance models
- simulation of Id/Vds with Vgs=0.0V
- repeat simulation using drift-diffusion models to compare results.

The example file consists of two ATLAS runs. Both use the same device structure defined in a similar manner to the previous example. The first uses energy balance models to simulate Id/Vds and the second does the same electrical simulation using the classical drift-diffusion models. The aim of this example is to compare the effect of the different models.

Energy balance models provide a more accurate description of physical device effects, in particular the effect of velocity overshoot and non-local impact ionization. These are not handled by the classical drift-diffusion model. In sub-micron MESFET simulation these two effects can be observed. This example concentrates on the velocity overshoot effect in GaAs. Non-local impact ionization effects can be seen in breakdown simulations. Since velocity overshoot is not accounted for in drift-diffusion the results from simulations with this model will underestimate the current severely. This discrepancy gets worse as the channel length decreases.

The sequence of simulation syntax is similar to the previous example. The structure differs from the previous only in the addition of heavily doped source/drain regions.

The energy balance models are defined using `model hcte.el`. Since this is a unipolar device only electron energy balance is selected. A key parameter in the energy balance model is the relaxation time. Typical GaAs values are set on the `material` statement.

A choice of numerical methods exists for energy balance simulation. Here, a full coupling of the four equations to be solved is selected by `method newton`. If convergence problems at the initial stage of the simulation are seen, the 'block' method may also be added.

The second run repeats the first but without `models hcte.el`. Results from the two runs can be compared by overlaying the two log files in TONYPLOT. Two solution files are also saved and these can be compared to see the difference in simulated electron concentration between the two models. The statement, `output e.velocity`, was used in both runs to save the electron velocity information to the solution files. Selecting to plot this for the energy balance solution file will show the overshoot effect.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

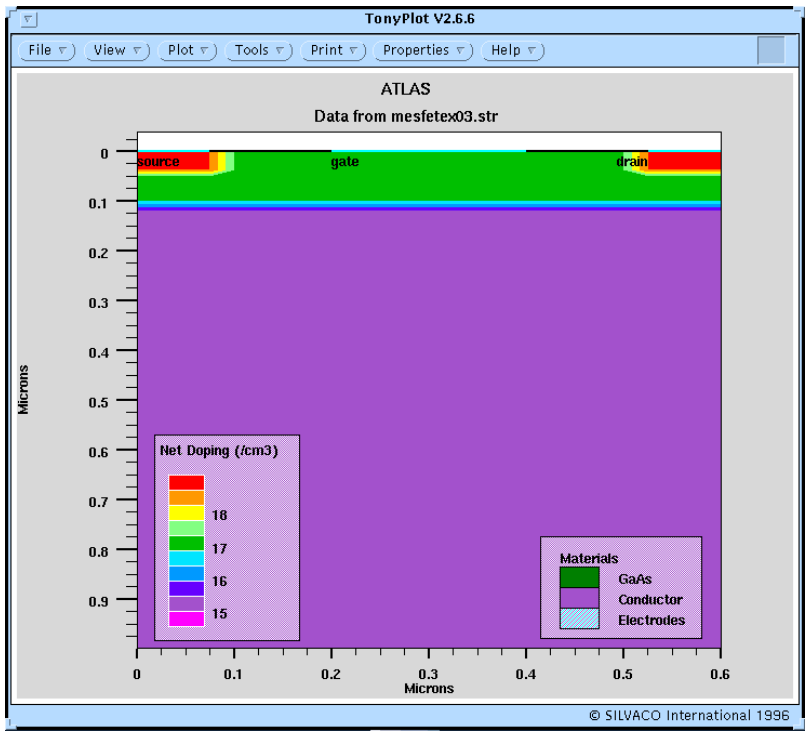


Figure 11.7: Doping profile of implanted MESFET defined using ATLAS syntax

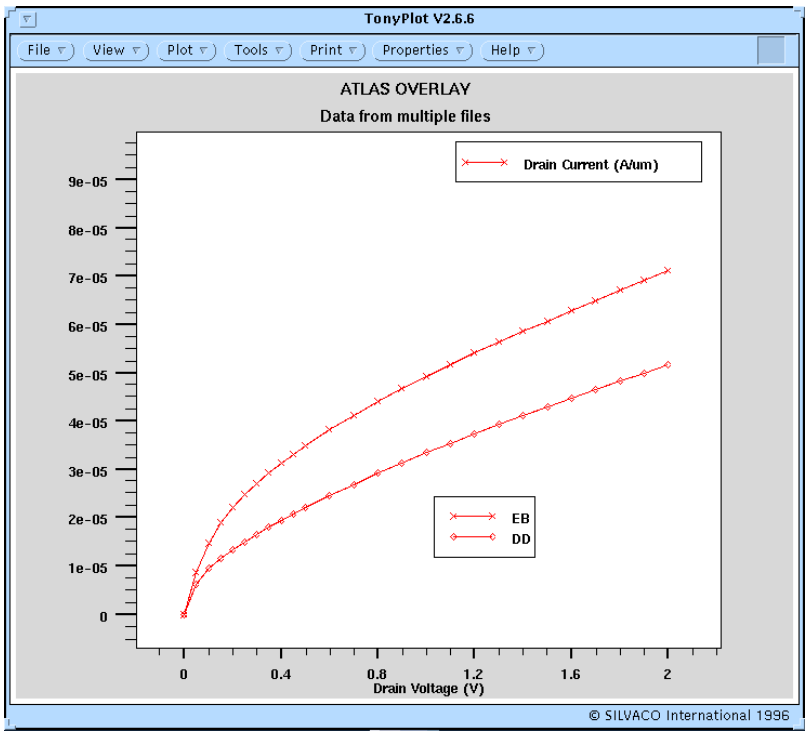


Figure 11.8: Comparison of energy balance and drift diffusion models for the MESFET Id/Vd curve. Energy balance models are required to simulate velocity overshoot effects

Input File mesfet/mesfetex03.in:

```
1  go atlas
2  title GaAs MESFET Simulation with EB and DD models
3  #
4  # SILVACO International 1996
5  #
6  # SECTION 1: Mesh spectification
7  #
8  # Define the mesh
9  #
10 mesh  space.mult=2.0
11 #
12 x.mesh loc=0.00 spac=0.02
13 x.mesh loc=0.2 spac=0.01
14 x.mesh loc=0.4 spac=0.01
15 x.mesh loc=0.6 spac=0.02
16 #
17 y.mesh loc=0.00 spac=0.005
18 y.mesh loc=0.2 spac=0.02
19 y.mesh loc=1 spac=0.1
20 #
21 # SECTION 2: Structure and models spectifications
22 #
23 region     num=1  material=GaAs
24 #
25 elec       num=1  name=source x.min=0.0 y.min=0.0 x.max=0.1 y.max=0.
26 elec       num=2  name=drain  x.min=0.5 y.min=0.0 x.max=0.6 y.max=0.
27 elec       num=3  name=gate   x.min=0.2 length=0.2
28 #
29 doping uniform conc=1.e15 p.type
30 doping uniform conc=1.e17 n.type y.min=0 y.max=0.12
31 doping uniform conc=5.e18  n.type x.left=0.  x.right=0.1 y.min=0
     y.max=0.05
32 doping uniform conc=5.e18  n.type x.left=0.5 x.right=0.6 y.min=0
     y.max=0.05
33 #
34 contact num=3 work=4.87
35 #
36 ######################################
37 #            EB calculation          #
38 ######################################
39 #
40 models   print  conmob fldmob  hcte.el
41 material taurel.el=1.e-12 taumob.el=1.e-12 vsat=1.e7
```

```
#
# SECTION 3: Initial solution
#
solve init
save outf=mesfetex03.str
#
tonyplot  mesfetex03.str -set mesfetex03_0.set
#
# SECTION 4: Id-Vd characteristics
#
method gummel newton  maxtrap=6
#
output e.velocity
#
probe n.mob max dir=0  name="mux"
probe n.mob max dir=90 name="muy"
log outf=mesfetex03_1.log
#
solve vdrain=0.0 vstep=0.05 vfinal=0.5  name=drain
method newton
solve  vdrain=0.6 vstep=0.1 vfinal=2 name=drain
save outf=mesfetex03_1.str
#
##########################################
#               DD calculation           #
##########################################
#
go atlas
#
# SECTION 1: Reload same mesh and structure spectification
#
mesh   infile=mesfetex03.str
#
contact num=3 work=4.87
models   print  conmob fldmob
material taurel.el=1.e-12 taumob.el=1.e-12 vsat=1.e7
#
# SECTION 2: Id-Vd characteristics
#
method gummel newton maxtrap=6
#
output e.velocity
#
```

```
probe n.mob max dir=0  name="mux"
probe n.mob max dir=90 name="muy"
log outf=mesfetex03_2.log
#
solve vdrain=0.0 vstep=0.05 vfinal=0.5  name=drain
method newton
solve  vdrain=0.6 vstep=0.1 vfinal=2 name=drain
save outf=mesfetex03_2.str
#
tonyplot  mesfetex03_1.log -overlay mesfetex03_2.log  -set
  mesfetex03_log.set
#
quit
```

11.1.4 mesfetex04.in: Deep Level Bulk Traps (EL2) - DC Analysis

Requires: FLASH/BLAZE

These next three examples demonstrate the simulation of a MESFET structure with trap states. The example consists of several sections:

- formation of a MESFET structure and doping using FLASH
- DC simulation of Id/Vgs characteristics including traps
- transient simulation of gate turn off including traps
- frequency domain AC simulation including traps
- s-parameter extraction including traps

This example creates the structure for subsequent usage and performs the DC analysis.

In the process simulation section, a simple planar MESFET is formed by implantation. The initial substrate is intrinsic GaAs. A low doping level of 1.0e11 is specified. An active layer of 0.1um of n-type GaAs is deposited. A nitride hard mask is used to pattern the source and drain regions. These regions are heavily doped using a silicon implant. Finally, metal deposition and patterning is done. The electrode names and positions are then defined at the final stage of ATHENA.

Once in ATLAS each run starts by setting the workfunction of the gate electrode. The models used in this simulation are electric field dependent mobility and SRH recombination. The lifetimes for the SRH recombination are set by the tau parameters on the material statement. The vsat parameter sets the saturation velocity.

The trap statement is used to set the parameter of the bulk trap states. Only one state is used here but several trap statements can be used in the same run to define multiple states. The trap type must be set as donor or acceptor. The energy level of the trap is set using e.level. The energy levels are referenced to the conduction or valence band edges. The sign and sigp parameters set the trapping cross sections for electrons and holes. An equivalent syntax setting lifetimes rather than cross sections is also available.

The DC simulation consists of setting the drain voltage to 0.5V and stepping the gate from zero to -1.0V. Comparing the results of this simulation with and without traps shows a shift to a more positive threshold voltage is caused by the presence of the donor traps.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

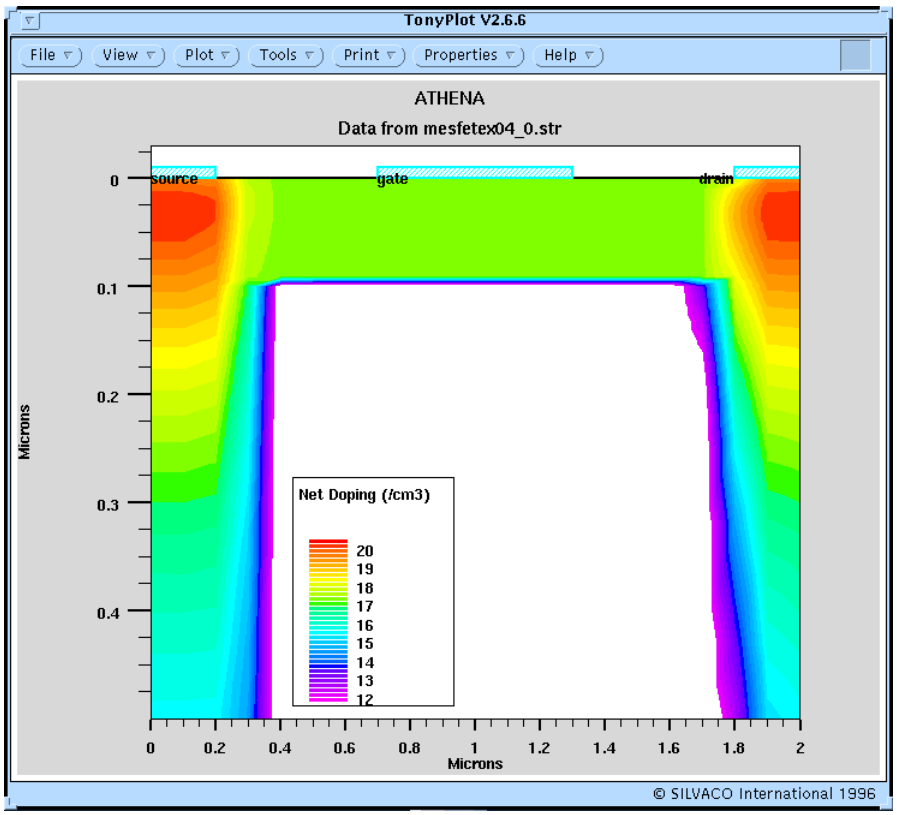


Figure 11.9: Doping profile of implanted MESFET defined in FLASH

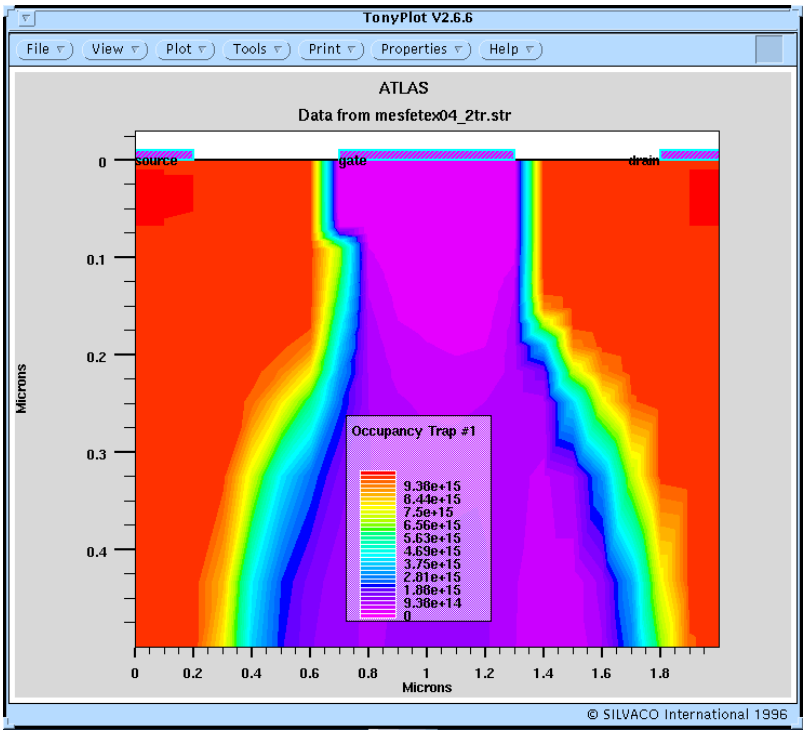


Figure 11.10: Occupancy of EL2 Traps in the MESFET during biasing

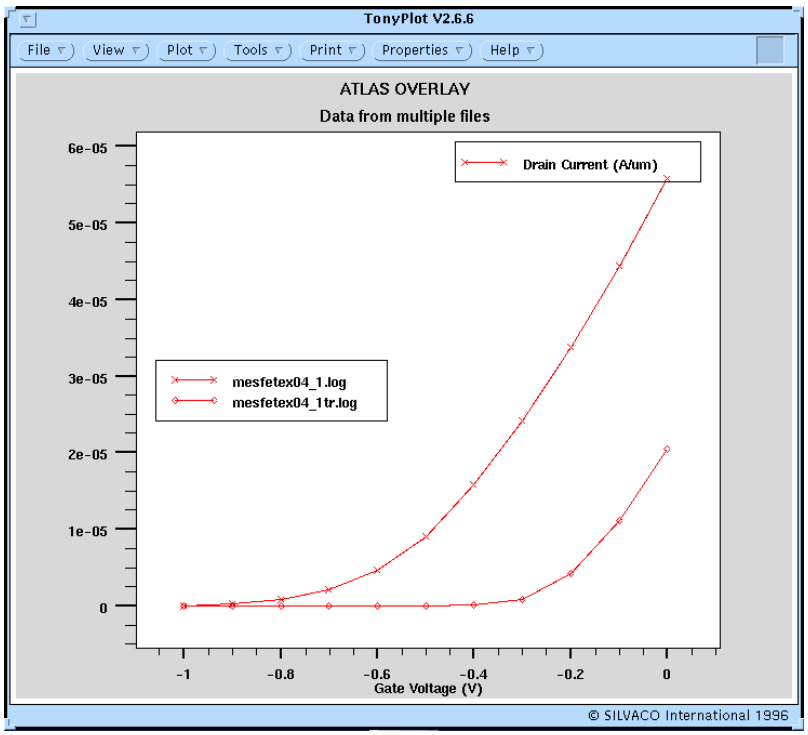


Figure 11.11: Comparison of MESFET threshold voltage using EL2 traps (o) and without traps (x)

Input File mesfet/mesfetex04.in:

```
1   go athena
2   #
3   line x loc=0.00 spac=0.10
4   line x loc=2 spac=0.10
5   #
6   line y loc=0.1 spac=0.01
7   line y loc=0.5 spac=0.1
8   #
9   init gaas c.silicon=1.0e11
10  #
11  deposit gaas thick=0.10 c.silicon=2.0e17 divisions=15 dy=0.005 ydy=0.00
12  #
13  deposit nitride thick=0.5 divisions=4
14  etch nitride left  p1.x=0.2
15  etch nitride right p1.x=1.8
16  relax y.min=0.2
17  #
18  implant silicon dose=1.0e15 energy=30
19  etch nitride all
20  #
21  deposit alum thickness=0.01
22  etch alum start x=0.2 y=-5
23  etch cont       x=0.7 y=-5
24  etch cont       x=0.7 y=5
25  etch done       x=0.2 y=5
26  etch alum start x=1.3 y=-5
27  etch cont       x=1.8 y=-5
28  etch cont       x=1.8 y=5
29  etch done       x=1.3 y=5
30  #
31  electrode  name=source x=0.1
32  electrode  name=gate x=1.0
33  electrode  name=drain x=1.9
34  #
35  structure outf=mesfetex04_0.str
36  tonyplot  mesfetex04_0.str -set mesfetex04_0.set
37
38  go atlas
39  #
40  contact name=gate work=4.87
41  models  print fldmob srh
42  #
```

```
material vsat=0.92e7 taun0=1e-9 taup0=1e-9
#
trap acceptor e.level=0.45 density=1.0e16 degen=12  sign=1e-14 sigp=3.e-
  14 fast
#
method newton trap
solve prev
solve vdrain=0.01
solve vdrain=0.05
solve vdrain=0.2
solve vdrain=0.5
#
log outf=mesfetex04_1tr.log master
solve vgate=0.0 vstep=-0.1 vfinal=-1.0 name=gate
save outf=mesfetex04_2tr.str
tonyplot  mesfetex04_2tr.str -set mesfetex04_1.set

go atlas
contact name=gate work=4.87
models  print fldmob srh
#
material vsat=0.92e7 taun0=1e-9 taup0=1e-9
#
method  newton trap
solve prev
solve vdrain=0.01
solve vdrain=0.05
solve vdrain=0.2
solve vdrain=0.5
#
log outf=mesfetex04_1.log
solve vgate=0.0 vstep=-0.1 vfinal=-1.0 name=gate
save outf=mesfetex04_2.str
tonyplot  mesfetex04_1.log -overlay mesfetex04_1tr.log
```

11.1.5 mesfetex05.in: Deep Level Bulk Traps (EL2) - Transient Analysis

Requires: FLASH/BLAZE

This example is based on the structure created in example 4 within the MESFET section - the trap simulation is now performed in transient mode.

In the transient turn-off simulation, the drain voltage is also set to 0.5V. However, the gate is then ramped in a transient from zero to-2.0V. The ramptime parameter sets the time for the gate voltage ramp, while tstop sets the time until which the simulation will continue. The parameter dt is used to set the first timestep of the transient. All other timesteps are calculated automatically by the pro-

gram. A dt value of ramptime/100 is safe one under most circumstances. Comparing the results of turn-off with and without traps shows how the presence of the fast trap states slows the time taken for the drain current to reduce to 1pA/um.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

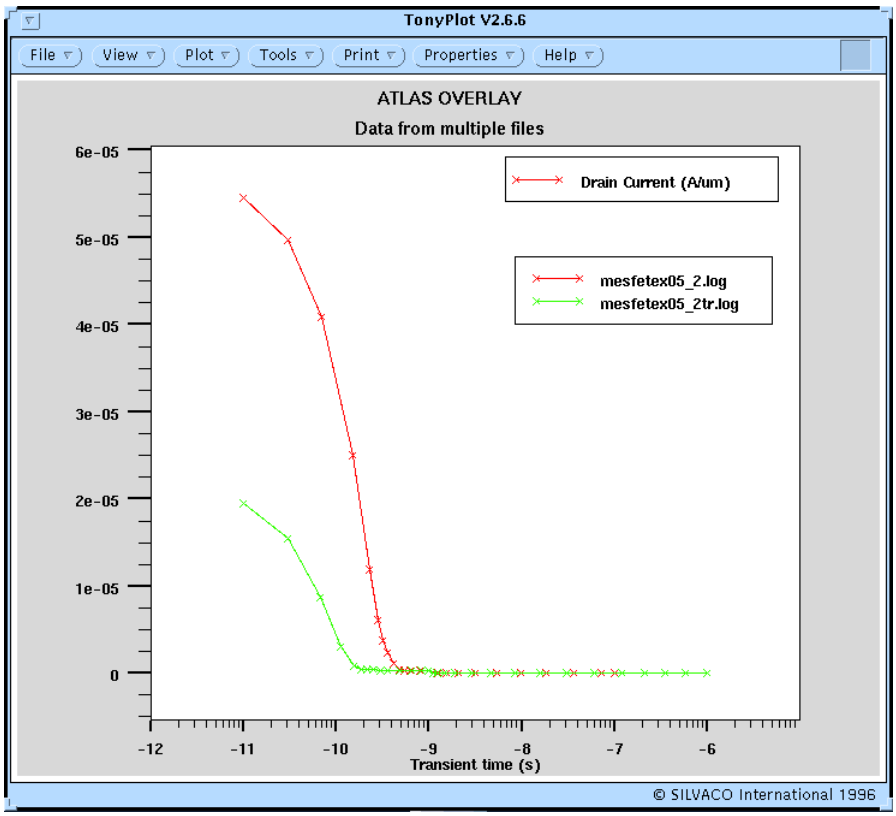


Figure 11.12: Effect of EL2 traps on transient switching of a MESFET. Traps (lower curve) and without traps (upper curve)

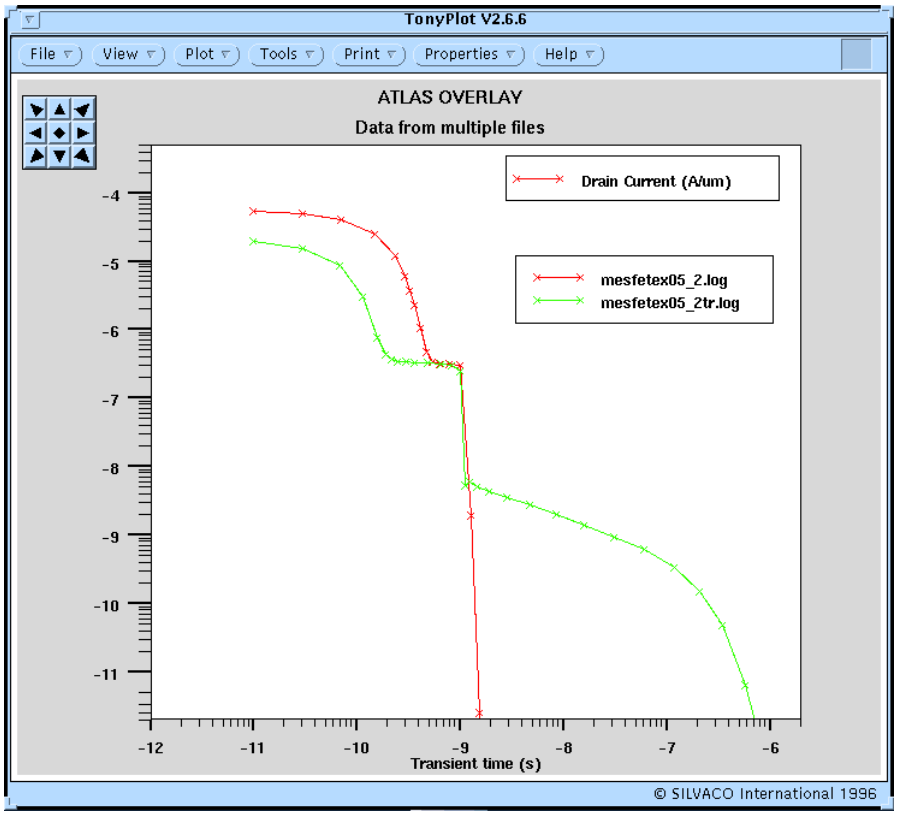


Figure 11.13: Effect of EL2 traps on transient switching of a MESFET using a log scale. Traps (lower curve extending to the right) and without traps (upper curve falling quickly to zero)

Input File mesfet/mesfetex05.in:

```
1   go athena
2   #
3   line x loc=0.00 spac=0.10
4   line x loc=2 spac=0.10
5   #
6   line y loc=0.1 spac=0.01
7   line y loc=0.5 spac=0.1
8   #
9   init gaas c.silicon=1.0e11
10  #
11  deposit gaas thick=0.10 c.silicon=2.0e17 divisions=15 dy=0.005 ydy=0.00
12  #
13  deposit nitride thick=0.5 divisions=4
14  etch nitride left  p1.x=0.2
15  etch nitride right p1.x=1.8
16  relax y.min=0.2
17  #
18  implant silicon dose=1.0e15 energy=30
```

```
19 etch nitride all
20 #
21 deposit alum thickness=0.01
22 etch alum start x=0.2 y=-5
23 etch cont       x=0.7 y=-5
24 etch cont       x=0.7 y=5
25 etch done       x=0.2 y=5
26 etch alum start x=1.3 y=-5
27 etch cont       x=1.8 y=-5
28 etch cont       x=1.8 y=5
29 etch done       x=1.3 y=5
30 #
31 electrode  name=source x=0.1
32 electrode  name=gate x=1.0
33 electrode  name=drain x=1.9
34 #
35 structure outf=mesfetex05_0.str
36 tonyplot  mesfetex05_0.str -set mesfetex05_0.set
37
38
39 go atlas
40 #
41 contact name=gate work=4.87
42 models  print fldmob srh
43 material vsat=0.92e7 taun0=1e-9 taup0=1e-9
44 trap acceptor e.level=0.45 density=1.0e16 degen=12  sign=1e-14 sigp=3.e-
     14 fast
45 #
46 method  newton trap
47 solve prev
48 solve vdrain=0.01
49 solve vdrain=0.05
50 solve vdrain=0.2
51 solve vdrain=0.5
52 #
53 log outf=mesfetex05_2tr.log
54 solve vgate=-2.0 ramptime=1e-9 dt=1e-11 tstop=1000e-9
55
56
57 go atlas
58 contact name=gate work=4.87
59 models  print fldmob srh
60 material vsat=0.92e7 taun0=1e-9 taup0=1e-9
```

```
#
method  newton trap
solve prev
solve vdrain=0.01
solve vdrain=0.05
solve vdrain=0.2
solve vdrain=0.5
#
log outf=mesfetex05_2.log
solve vgate=-2.0 ramptime=1e-9 dt=1e-11 tstop=100e-9
tonyplot  mesfetex05_2.log -overlay mesfetex05_2tr.log -set
  mesfetex05_1.set

```

11.1.6 mesfetex06.in: Deep Level Bulk Traps (EL2) - AC Analysis

Requires: FLASH/BLAZE

This example is based on the structure created in example four within the MESFET section - the trap simulation is now performed in ac mode.

The AC simulation uses the same initial point as the previous two cases. The gate is grounded and the drain is at 0.5V. An AC signal is applied at this point to each contact in turn. The frequency of this AC signal is then ramped from 1MHz to over 10GHz. The `fstep` parameter sets the step size of the frequency change. However the parameter `mult.f` means `fstep` is applied as a multiplier to the frequency rather than as a sum.

As described in example two in this section, the `s.param` command on the log statement is used to specify the output of s-parameters.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

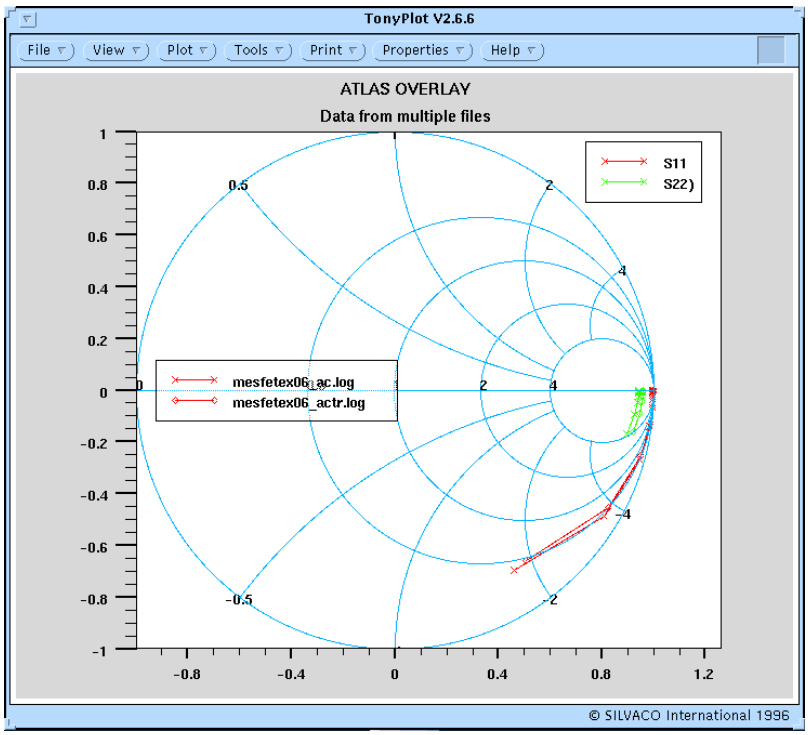


Figure 11.14: Effect of EL2 traps on S11 and S22 parameters. Traps (o) and No traps (x)

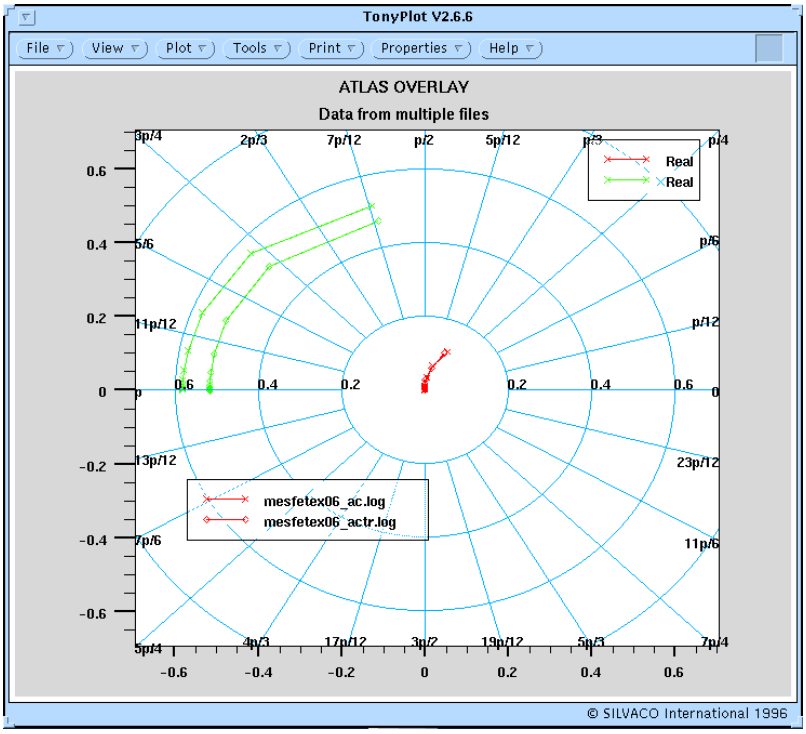


Figure 11.15: Effect of EL2 traps on S12 and S21 parameters. Traps (o) and No traps (x)

Input File mesfet/mesfetex06.in:

```
1   go athena
2   #
3   line x loc=0.00 spac=0.10
4   line x loc=2 spac=0.10
5   #
6   line y loc=0.1 spac=0.01
7   line y loc=0.5 spac=0.1
8   #
9   init gaas c.silicon=1.0e11
10  #
11  deposit gaas thick=0.10 c.silicon=2.0e17 divisions=15 dy=0.005 ydy=0.00
12  #
13  deposit nitride thick=0.5 divisions=4
14  etch nitride left  p1.x=0.2
15  etch nitride right p1.x=1.8
16  relax y.min=0.2
17  #
18  implant silicon dose=1.0e15 energy=30
19  etch nitride all
20  #
21  deposit alum thickness=0.01
22  etch alum start x=0.2 y=-5
23  etch cont       x=0.7 y=-5
24  etch cont       x=0.7 y=5
25  etch done       x=0.2 y=5
26  etch alum start x=1.3 y=-5
27  etch cont       x=1.8 y=-5
28  etch cont       x=1.8 y=5
29  etch done       x=1.3 y=5
30  #
31  electrode  name=source x=0.1
32  electrode  name=gate x=1.0
33  electrode  name=drain x=1.9
34  #
35  structure outf=mesfetex06_0.str
36  tonyplot  mesfetex06_0.str -set mesfetex06_0.set
37
38
39  go atlas
40  #
41  contact name=gate work=4.87
42  models  print fldmob srh
```

```
material vsat=0.92e7 taun0=1e-9 taup0=1e-9
trap acceptor e.level=0.45 density=1.0e16 degen=12  sign=1e-14 sigp=3.e-
  14 fast
#
method newton trap
solve prev
solve vdrain=0.01
solve vdrain=0.05
solve vdrain=0.2
solve vdrain=0.5
#
log outf=mesfetex06_actr.log s.param inport=gate outport=drain width=50
  imped=50
solve prev ac freq=1e6 mult.f fstep=2 nfstep=15

go atlas
contact name=gate work=4.87
models  print fldmob srh
material vsat=0.92e7 taun0=1e-9 taup0=1e-9
#
method  newton trap
solve prev
solve vdrain=0.01
solve vdrain=0.05
solve vdrain=0.2
solve vdrain=0.5
#
log outf=mesfetex06_ac.log s.param inport=gate outport=drain width=50
  imped=50
solve prev ac freq=1e6 mult.f fstep=2 nfstep=15
#
tonyplot  mesfetex06_ac.log -overlay mesfetex06_actr.log -set
  mesfetex06_1.set
tonyplot  mesfetex06_ac.log -overlay mesfetex06_actr.log -set
  mesfetex06_2.set
#
quit


```

12.1. HBT: HBT Application Examples

12.1.1 hbtex01.in: Si/SiGe HBT Gummel Plot Simulation

Requires: BLAZE

This example demonstrates the simulation of a SiGe HBT to extract Ic and Ib versus Vbe and plots the bipolar gain. It shows:

- device formation using ATLAS syntax
- specification of a graded SiGe composition fraction
- selection of models for SiGe
- DC simulation for Ic and Ib versus Vbe

The majority of this example is similar to the silicon bipolar Gummel plot example described in the BJT section. This description will focus on the specific SiGe syntax. More details on various types of bipolar simulation can be found in the BJT examples section.

The `region` statement is used to define the SiGe region in the base of an NPN bipolar transistor. The argument of the 'material' parameter is case-sensitive: SiGe is acceptable, but sige is not. The composition fraction of the area inside of x.min, x.max, y.min and y.max is defined by the `x.composition` parameter. Outside of this box the Ge composition fraction rolls off linearly to zero over the distance specified by the `grad.*` parameters. For more details on `grad.*` parameters, see the BLAZE chapter of the manual. For non-linear composition fraction variations, DEVEDIT is needed.

The material parameters for SiGe are selected by `material material=SiGe ....` Here the lifetime is set differently for SiGe and Si regions. In this example common models are used between Si and SiGe regions, but this is not compulsory. The material parameter of the models statement can be used to set separate models for each material.

The electrical part of the example is the same as the silicon examples referenced above. The solution file stored at the final voltage can be used to examine the band diagram of the HBT. The statement:

```
output con.band val.band
```

enables the conduction and valence band edge energies to be stored in the solution file. This command is recommended for all heterojunction simulations in BLAZE.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

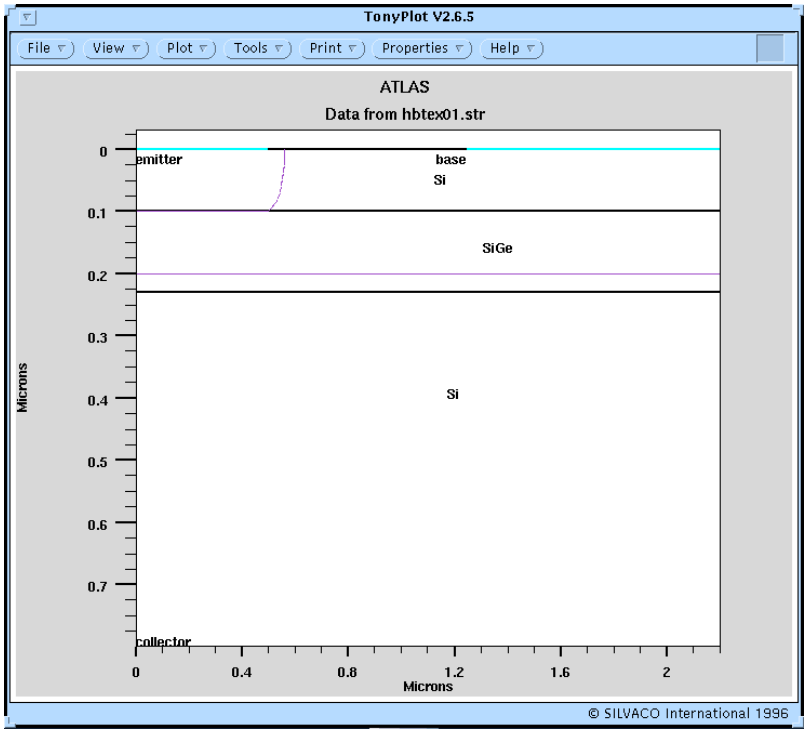


Figure 12.1: Heterojunction and dopant junction locations for a SiGe HBT defined in ATLAS syntax

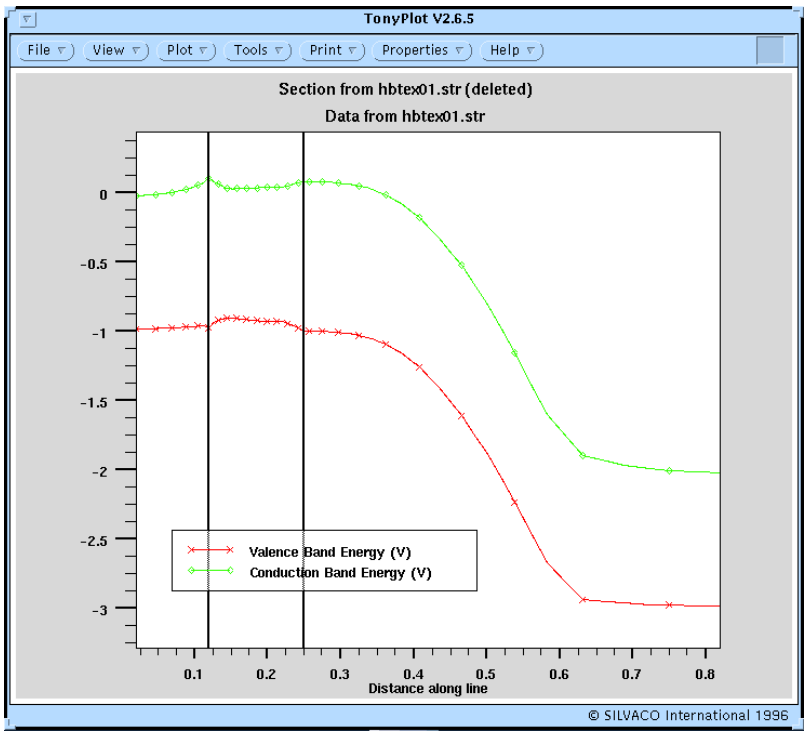


Figure 12.2: Band Diagram of the HBT with applied Vce

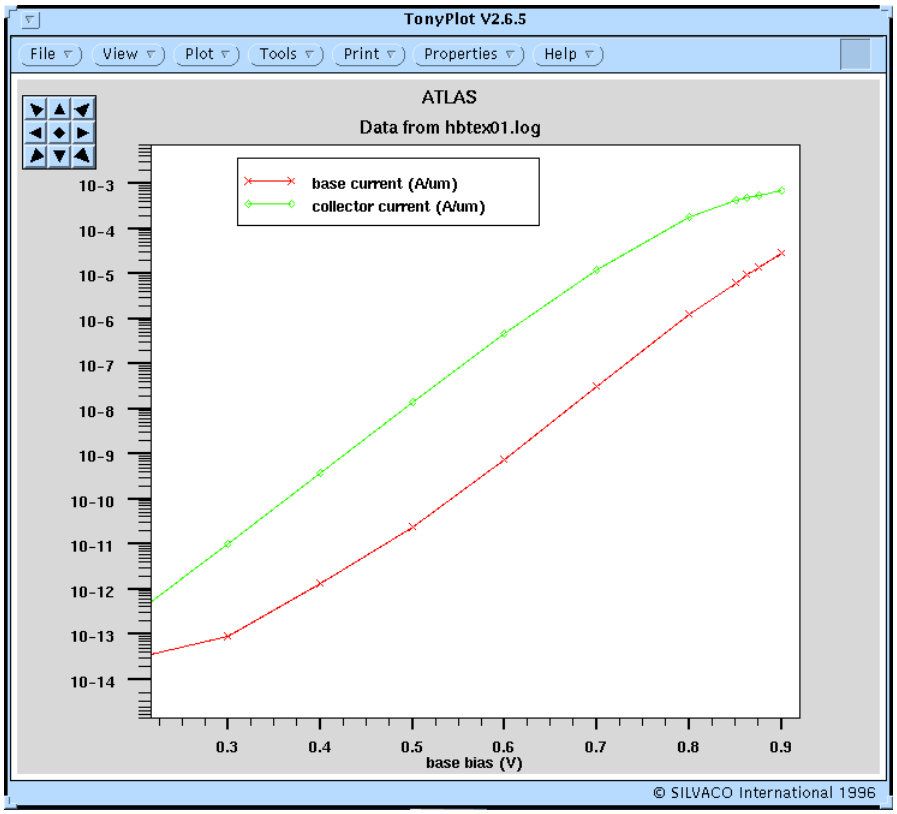


Figure 12.3: Gummel plot of the SiGe HBT

Input File hbt/hbtex01.in :

```
1   go atlas
2   title SiGe HBT simulation
3   #
4   # SILVACO International 1992, 1993, 1994
5   #
6   # SECTION 1: Mesh spectification
7   #
8   mesh space.mult=0.7
9   #
10  x.m  loc=0.0   spacing=0.1
11  x.m  loc=0.5   spacing=0.05
12  x.m  loc=0.75  spacing=0.1
13  x.m  loc=2.2   spacing=0.2
14  #
15  y.m  loc=0.0   spacing=0.02
16  y.m  loc=0.10  spacing=0.01
17  y.m  loc=0.125 spacing=0.01
18  y.m  loc=0.2   spacing=0.01
19  y.m  loc=0.23  spacing=0.01
```

```
y.m  loc=0.8   spacing=0.1
#
# SECTION 2: Structure definition
#
region   num=1 material=Si   y.min=0.2
region   num=2 material=Si   y.max=0.1250
region   num=3 material=SiGe y.min=0.1250 y.max=0.2 x.composition=0.2
  grad.1=0.0250 grad.3=0.03
#
elec    name=emitter x.min=0.0 x.max=0.5 y.min=0.0 y.max=0.0
elec    name=base    x.min=1.2 x.max=2.2 y.min=0.0 y.max=0.0
elec    name=collector bot
#
doping uniform  n.type conc=2.e16
doping gauss    n.type conc=8.e19 peak=0.8  char=0.090
doping gauss    p.type conc=3.e19 junc=0.200
doping gauss    n.type conc=1.e20 junc=0.1  x.right=0.5 lat.char=0.05
doping gauss    p.type conc=5.e19 peak=0.0  char=0.05  x.left=1.2
  lat.char=0.2
#
# SECTION 3: Material model specification
#
material material=Si    taun0=1e-7 taup0=1e-7
material material=SiGe  taun0=1.e-8 taup0=1.e-8
model    bgn  srh  auger fldmob conmob

save outf=hbtex01_0.str

#
# SECTION 4: Initial solution
#


solve init
#

method gummel newton trap autonr
solve prev
solve local vcollector=2.0
#
# SECTION 5: Gummel plot
#
output con.band val.band
log outf=hbtex01.log master
```

```
62 #
63
64 method  newton  trap
65 solver vbase=0.01
66 solver vbase=0.05
67 solve vbase=0.1 vstep=0.1 name=base vfinal=0.9
68
69 save outf=hbtex01.str
70 tonyplot  hbtex01.str -set hbtex01str.set
71 tonyplot  hbtex01.log -set hbtex01gum.set
72
73 quit
```

12.1.2 hbtex02.in: AlGaAs/GaAs HBT Gummel Plot Simulation

Requires: BLAZE

This example demonstrates the simulation of an AlGaAs/GaAs HBT to extract Ic and Ib versus Vbe and plots the bipolar gain. It shows:

- device formation using ATLAS syntax
- specification of s graded AlGaAs composition fraction
- selection of models for AlGaAs and GaAs
- DC simulation for Ic and Ib versus Vbe

As in the previous example, the majority of this example is similar to the silicon bipolar Gummel plot example described in the BJT section. This description will focus on specific GaAs and AlGaAs syntax. More details on various types of bipolar simulation can be found in the BJT examples section.

The graded AlGaAs composition fraction is set in the `region` statement. The parameter, `x.comp`, defines the Al content inside the box defined by `x.min`, `x.max`, `y.min` and `y.max`. Outside the box the Al content falls linearly to zero in a distance `grad.*`. For more details on `grad.*` parameters see the BLAZE chapter of the manual. Non-linear composition gradings can only be handled by DEVEDIT. Since devices constructed in ATLAS syntax must be rectangular a silicon dioxide region is used to fill the space to the left of the emitter.

The `material` statement sets the low-field mobilities for the AlGaAs region. Note that the concentration dependent mobility model 'conmob' can only be applied to GaAs so two models statements are used to define the models used in all regions and then just the GaAs regions.

The `model` statements are used to specify the following set of models: field dependent mobility, SRH and optical recombination. Optical recombination (also known as band to band recombination) is an important effect in III-V devices. The parameter, `optr`, sets this model on.

The electrical simulation proceeds as in the silicon BJT example referenced above. Results can be seem in TONYPLOT.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

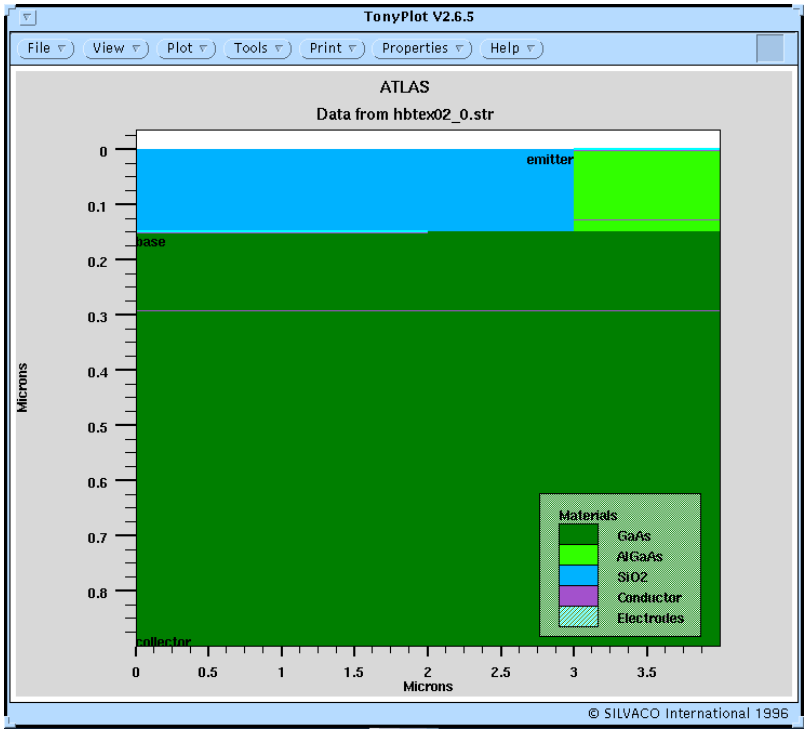


Figure 12.4: Structure of AlGaAs/GaAs HBT defined using ATLAS syntax

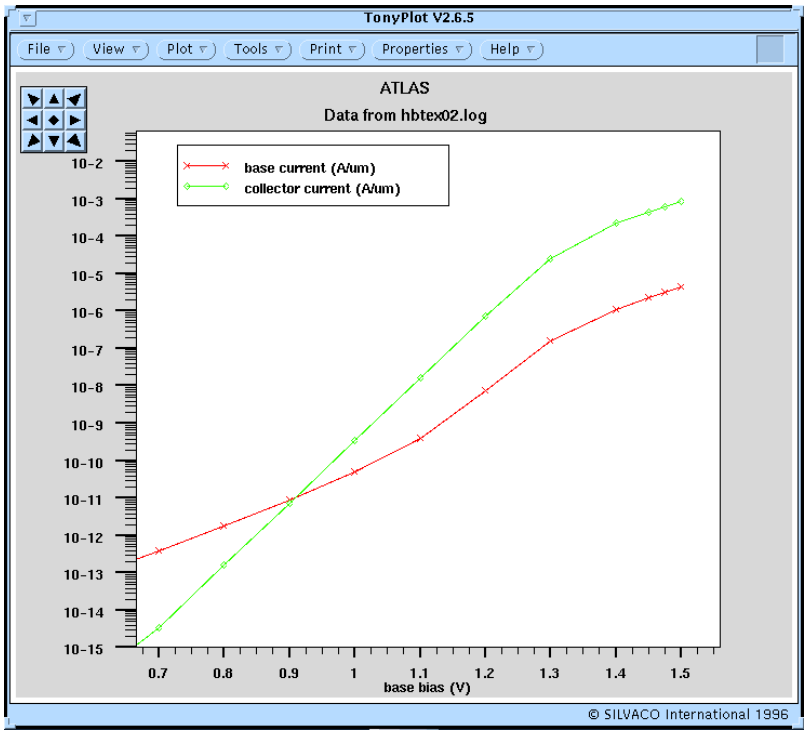


Figure 12.5: Gummel Plot for AlGaAs/GaAs HBT

Input File hbt/hbtex02.in:

```
1   go atlas
2   title AlGaAs HBT simulation with DD model
3   #
4
5   # SILVACO International 1993, 1994
6   #
7   # SECTION 1: Mesh spectification
8   #
9   mesh       nx=24 ny=24
10  x.m        n=1    l=0.0   r=1.
11  x.m        n=10   l=2.0   r=1.0
12  x.m        n=18   l=3.0   r=1.0
13  x.m        n=24   l=4.0   r=1.0
14  #
15  y.m        n=1    l=0.0   r=1.0
16  y.m        n=6    l=0.15  r=0.9
17  y.m        n=12   l=0.25  r=0.9
18  y.m        n=20   l=0.75  r=1.1
19  y.m        n=24   l=0.9   r=1.0
20  #
21  eliminate x.dir iy.low=1 iy.high=5 ix.low=1 ix.high=17
22  #
23  # SECTION 2: Structure definition
24  #
25  region num=1 material=GaAs y.min=0.75
26  region num=2 material=GaAs y.min=0.25 y.max=0.75
27  region num=3 material=AlGaAs y.max=0.0 x.comp=0.3 \
28   grad.34=0.15
29  region num=4 material=GaAs y.min=0.15 y.max=0.25
30  region num=5 material=SiO2 y.max=0.15 x.max=3.0
31  #
32  elec num=1  name=emitter x.min=3.0 x.max=4.0 y.min=0.0 y.max=0.0
33  elec num=2  name=base    x.min=0.0 x.max=2.0 y.min=0.15 y.max=0.15
34  elec num=3  name=collector bot
35  #
36  doping uniform region=1 n.type conc=2.e17
37  doping uniform region=2 n.type conc=5.e16
38  doping uniform region=3 n.type conc=2.e17
39  doping uniform region=4 p.type conc=2.e18
40
41  save outf=hbtex02_0.str
42  tonyplot  hbtex02_0.str -set hbtex02_0.set
```

```
43
44 #
45 #
46 # SECTION 3: Set models and define material parameters
47 #
48 material taun0=1.e-9 taup0=1.e-9
49 material material=AlGaAs mun=2200 mup=350
50
51 model  srh fldmob   print  optr
52 model material=GaAs conmob evsatmod=1
53 #
54 # SECTION 4: Initial solution and collector bias ramp
55 solve init
56
57
58 method gummel  newton trap  vsatmod.inc=0.01
59
60 solve
61 solve local vcollector=2.0
62
63 #
64 # SECTION 5: Calculations for Gummel plot
65 #
66
67 log outf=hbtex02.log master
68
69
70 solve vbase=0.025
71 solve vbase=0.1
72
73 method  newton  trap autonr vsatmod.inc=0.01 itlim=50
74
75 solve vbase=0.2 vstep=0.1 name=base vfinal=1.4
76 solve vbase=1.45 vstep=0.025 name=base vfinal=1.5
77
78
79 tonyplot hbtex02.log -set hbtex02.set
80
81 quit
```

12.1.3 hbtex03.in: AlGaAs/GaAs HBT Avalanche Breakdown

Requires: BLAZE

This example demonstrates avalanche breakdown in an AlGaAs/GaAs HBT with an **abrupt heterojunction.** It shows:

- device formation using ATLAS syntax
- specification of an abrupt AlGaAs/GaAs heterojunction
- selection of models for AlGaAs and GaAs
- ramping of the collector with an applied resistor until the breakdown voltage

The structure specification part of this example is similar to the previous example. The major difference is that this HBT is an abrupt AlGaAs/GaAs heterojunction as opposed to the graded composition fraction used before. Specifying an abrupt junction is simple. No grad.* parameters are used and the value of `x.comp` on the `region` statement is taken as the composition fraction throughout the AlGaAs region right up to the GaAs region.

The models and material parameters are set exactly as in the previous example. The only addition is the impact ionization model specified by `impact selb`. Impact ionization parameters for ternary compounds such as AlGaAs are not well defined and their dependence on composition fraction is known even less. Tuning of the parameters of the impact statement might be needed to match experimental results.

The `contact` statement is used to specify a high value resistor at the collector electrode. This provides for a smooth transition from voltage boundary conditions to current boundary conditions during the ramping of external collector voltage. Note that the same kind of simulation could be performed using the curve tracing algorithm, providing automatic selection of the loaded resistor.

The emitter voltage is ramped to -1.1V. Then by holding the emitter voltage constant, the external collector voltage is ramped to 1.e9 V. Due to the high value resisto, the value seen on the semiconductor-to-collector contact will, of course, not increase to such high levels.

The results of simulation are then displayed using TONYPLOT. The x-axis of the Ic/Vcb curve should be set to `collector int. bias.` This is the collector voltage on the semiconductor contact as opposed to the applied voltage across the resistor.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

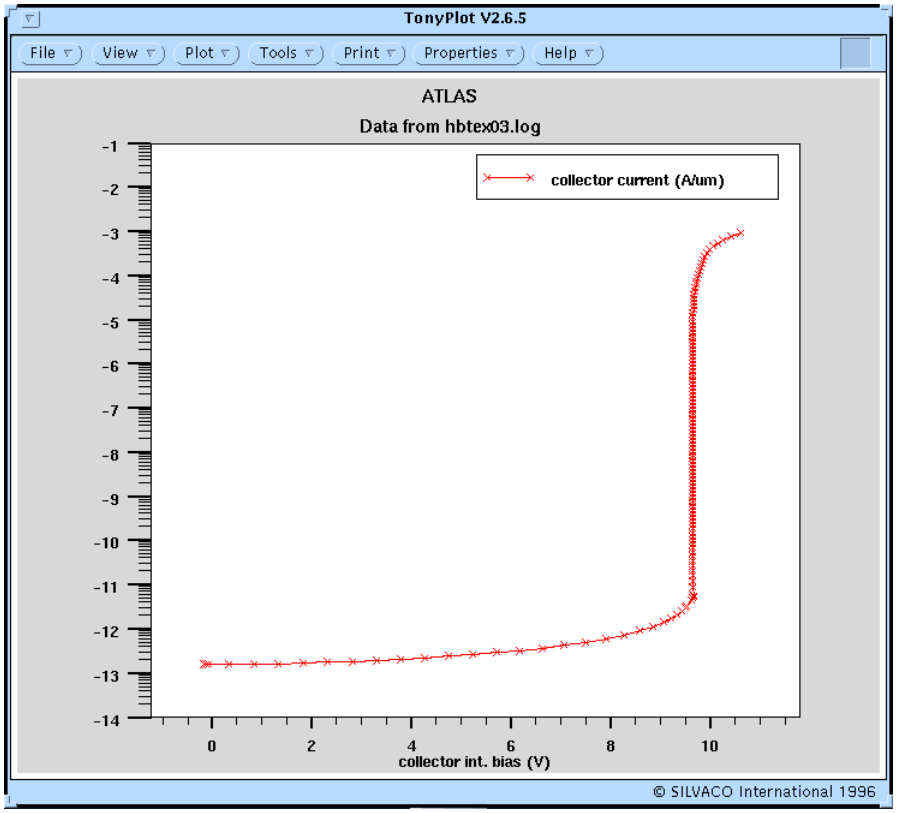


Figure 12.6: Breakdown Voltage Curve for AlGaAs/GaAs HBT

Input File hbt/hbtex03.in :

```
1   go atlas
2   title AlGaAs HBT breakdown simulation
3
4   # SILVACO International 1993, 1994
5   #
6   # SECTION 1: Mesh specification
7   #
8   mesh       nx=18 ny=41
9   x.m        n=1    l=0.0   r=1.
10  x.m        n=8   l=2.0   r=1.0
11  x.m        n=14   l=3.0   r=1.0
12  x.m        n=18   l=4.0   r=1.0
13  #
14  y.m        n=1    l=0.0   r=1.0
15  y.m        n=9    l=0.2   r=0.9
16  y.m        n=10   l=0.204 r=1.0
17  y.m        n=17   l=0.38  r=0.95
18  y.m        n=35   l=0.7  r=1.08
19  y.m        n=41   l=1.2   r=1.0
```

```
#
# SECTION 2: Structure definition
#
region num=1 material=GaAs y.min=0.7
region num=2 material=GaAs y.min=0.4 y.max=0.7
region num=3 material=AlGaAs y.max=0.2 x.comp=0.3
region num=4 material=GaAs y.min=0.2 y.max=0.4
region num=5 material=SiO2 y.max=0.2 x.max=3.0
#
elec num=1  name=emitter x.min=3.0 x.max=4.0 y.min=0.0 y.max=0.0
elec num=2  name=base    x.min=0.0 x.max=2.0 y.min=0.2 y.max=0.2
elec num=3  name=collector bot
#
doping uniform region=1 n.type conc=1.e18
doping uniform region=2 n.type conc=2.e17
doping uniform region=3 n.type conc=1.e18
doping uniform region=4 p.type conc=1.e19

save outf=hbtex03_0.str
tonyplot hbtex03_0.str -set hbtex03_0.set
#
material taun0=1.e-9 taup0=1.e-9
material material=AlGaAs mun=2000 mup=350
#
model srh fldmob print  conmob
#
model material=GaAs evsatmod=1

contact num=3 r=1.e12

impact selb


solve init


method  newton trap  itlim=25 vsatmod.inc=0.01 \
   ir.tol=1.e-20 ix.tol=1.e-20

solve local vemitter=-0.25
solve vemitter=-0.5 vstep=-0.2 vfinal=-1.1 name=emitter

output con.band val.band
```

```
63
64 log outf=hbtex03.log master
65
66 solve vcollector=0.0005
67 solve vcollector=0.005
68 solve vcollector=0.05
69 solve vcollector=0.1
70 solve vcollector=0.5 vstep=0.5 name=collector vfinal=12.5
71 solve vcollector=12.75 vstep=1.2 imult name=collector \
72  vfinal=1.e7
73 solve vcollector=1.2e7 vstep=1.2 imult name=collector \
74  vfinal=1.e9
75
76 tonyplot  hbtex03.log -set hbtex03.set
77
78 quit
79
```

12.1.4 hbtex04.in: Energy Balance and Non-Isothermal Energy Balance

Requires: BLAZE/GIGA

This example compares the results of Ic/Vce simulation of an AlGaAs/GaAs HBT using the Energy Balance (EB) and Nonisothermal Energy Balance (NEB) Models. It shows:

- Graded heterojunction AlGaAs/GaAs HBT formation using ATLAS syntax
- specification of energy balance models
- solution for Ic/Vce characteristics
- comparative simulation including lattice heat flow

Deep GaAs submicron devices should be simulated using the energy balance Model due to strong velocity overshoot, which may substantially influence device characteristics. For high current levels the thermal self-heating effects may also play an important role, due to the relatively small heat conductivity in GaAs, thereby decreasing mobility and leading to a negative output conductance. This example demonstrates a comparison of Ic-Vce curves obtained with Energy Balance and Nonisothermal Energy Balance Models.

Two ATLAS runs are included in this example. The initial stages of each uses ATLAS syntax to specify a graded heterojunction HBT. The syntax used is similar to the AlGaAs/GaAs HBT examples described before. The first run proceeds to select and solve self-consistently the classical equations: Poisson's equation and the two carrier continuity equations, and the energy balance equation for electrons. The second run adds the lattice heat flow equation to these four and solves self-consistently using a coupled solver.

The energy balance model is selected by `models hcte.el`. The most important parameter in energy balance simulation are the relaxation times. These are set on the `material` statement.

In each run the base bias is ramped to 1.2V, then `contact num=2 current` is used to set current boundary conditions on the base electrode. The base current is forced to a value of 15uA/um. After this the collector voltage is ramped to 2.0V to obtain the Ic/Vce curve.

The same set of models is used, except that the solution of the lattice energy balance equation is activated using the syntax, `models lat.temp`. For the AlGaAs region the temperature depen-

dence of the low field mobilities is set by the tmup and tmun parameters on the material statement.

As with all non-isothermal simulation, thermal boundary conditions must be included. These are defined in the thermcontact statement. A value of the thermal resistance is specified at the thermal contact located along the substrate, and thermal isolation conditions are assumed on the all other surfaces.

The results of the two runs can be overlaid in TONYPLOT. The simulation with lattice heating shows less current for a given voltage than the isothermal case. The elevated temperature causes mobility reduction in the semiconductor and hence lower currents.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

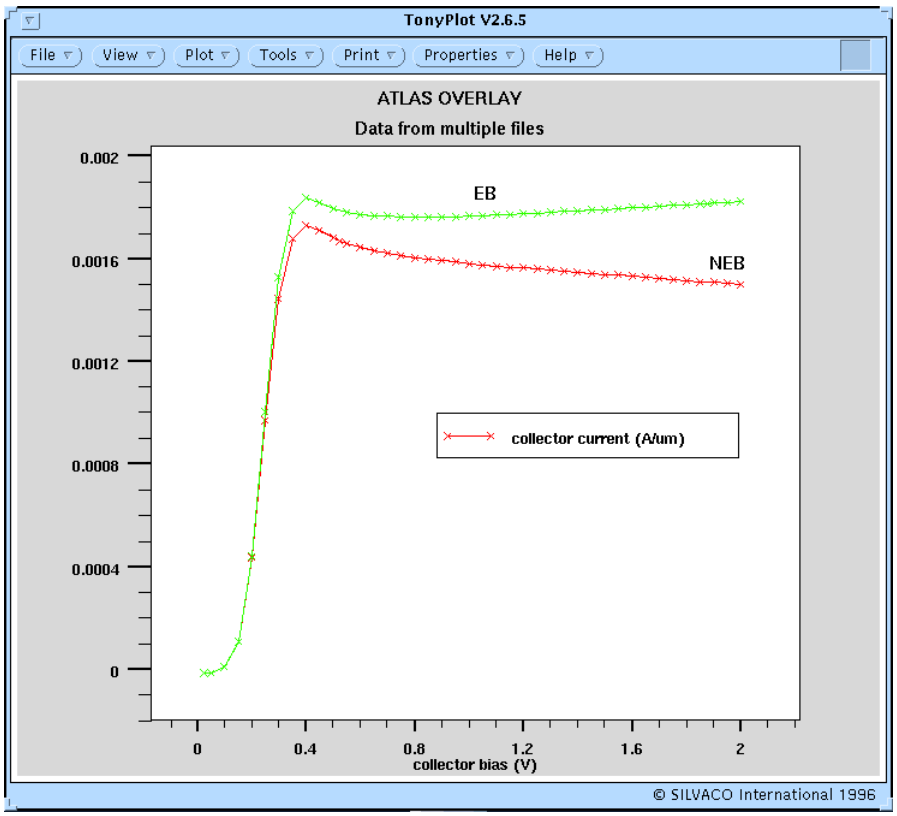


Figure 12.7: Comparison of isothermal and non-isothermal energy balance models in an AlGaAs/GaAs HBT. The negative conductance slope is typical of self-heating in devices

Input File hbt/hbtex04.in:

```
1  go atlas
2  TITLE AlGaAs/GaAs HBT Ic-Vce simulation with EB and NEB Models
3  # SILVACO International 1994
4
5  # EB simulation
6
7
8  # SECTION 1: Mesh spectification
```

```
9
10 mesh       rect  nx=16 ny=85
11 x.m        n=1    l=0.0   r=1.0
12 x.m        n=5   l=2.0    r=1.0
13 x.m        n=8   l=2.8    r=1.0
14 x.m        n=12   l=3.2   r=1.0
15 x.m        n=16   l=4.0   r=1.0
16 y.m        n=1    l=0.0   r=1.0
17 y.m        n=8    l=0.1   r=1.0
18 y.m        n=16   l=0.15  r=0.95
19 y.m        n=23   l=0.2   r=1.0
20 y.m        n=30   l=0.25  r=0.9
21 y.m        n=58   l=0.40  r=1.0
22 y.m        n=65   l=0.70  r=1.0
23 y.m        n=85   l=0.90  r=1.0
24
25 # SECTION 2: Structure definition
26
27 region num=1 material=GaAs y.min=0.75
28 region num=2 material=GaAs y.min=0.25 y.max=0.75
29 region num=3 material=AlGaAs y.max=0.12 x.comp=0.3 grad.34=0.03
30 region num=4 material=GaAs y.min=0.15 y.max=0.25
31 region num=5 material=SiO2 y.max=0.15 x.max=3.0
32
33 elec num=1  name=emitter x.min=3.0 x.max=4.0 y.min=0.0 y.max=0.0
34 elec num=2  name=base    x.min=0.0 x.max=2.0 y.min=0.15 y.max=0.15
35 elec num=3  name=collector bot
36
37 doping uniform region=1 n.type conc=1.e18
38 doping uniform region=2 n.type conc=1.e17
39 doping uniform region=3 n.type conc=5e17
40 doping uniform region=4 p.type conc=1.e19
41
42
43 save outf=hbtex04_0.str
44 tonyplot  hbtex04_0.str -set hbtex04_0.set
45
46
47
48 material taun0=2.e-10 taup0=2.e-10  taurel.el=1.e-12 \
49          taumob.el=1.e-12
50
51 material material=AlGaAs mun=2170 mup=350 tmun=1 tmup=2.1
```

```
52
53 model srh  fldmob  print hcte.el
54 model material=GaAs conmob evsatmod=1
55
56
57 output val.band con.band e.velocity
58
59
60 solve init
61
62 method  newton  maxtrap=6  temp.tol=1.e-4  trap itlim=50 \
63   vsatmod.inc=0.01
64
65 solve vbase=0.005
66 solve vbase=0.02
67 solve vbase=0.05
68 solve vbase=0.2 vstep=0.2 vfinal=1.2 electr=2
69
70 contact num=2 current
71 solve i2=1.e-5
72 solve i2=1.2e-5
73 solve i2=1.5e-5
74
75 log outf=hbtex04_eb.log master
76
77
78 solve vcollector=0.025
79 solve vcollector=0.05
80 solve vcollector=0.1 vstep=0.05 vfinal=2 electr=3
81
82 save outf=hbtex04_eb.str
83
84 go atlas
85
86 # NEB simulation
87
88
89 # SECTION 1: Mesh spectification
90
91 mesh       rect  nx=16 ny=85
92 x.m        n=1    l=0.0   r=1.0
93 x.m        n=5   l=2.0    r=1.0
94 x.m        n=8   l=2.8    r=1.0
```

```
x.m        n=12   l=3.2   r=1.0
x.m        n=16   l=4.0   r=1.0
y.m        n=1    l=0.0   r=1.0
y.m        n=8    l=0.1   r=1.0
y.m        n=16   l=0.15  r=0.95
y.m        n=23   l=0.2   r=1.0
y.m        n=30   l=0.25  r=0.9
y.m        n=58   l=0.40  r=1.0
y.m        n=65   l=0.70  r=1.0
y.m        n=85   l=0.90  r=1.0



# SECTION 2: Structure definition

region num=1 material=GaAs y.min=0.75
region num=2 material=GaAs y.min=0.25 y.max=0.75
region num=3 material=AlGaAs y.max=0.12 x.comp=0.3 grad.34=0.03
region num=4 material=GaAs y.min=0.15 y.max=0.25
region num=5 material=SiO2 y.max=0.15 x.max=3.0

elec num=1  name=emitter x.min=3.0 x.max=4.0 y.min=0.0 y.max=0.0
elec num=2  name=base    x.min=0.0 x.max=2.0 y.min=0.15 y.max=0.15
elec num=3  name=collector bot

doping uniform region=1 n.type conc=1.e18
doping uniform region=2 n.type conc=1.e17
doping uniform region=3 n.type conc=5e17
doping uniform region=4 p.type conc=1.e19



material taun0=2.e-10 taup0=2.e-10  taurel.el=1.e-12  \
         taumob.el=1.e-12

material material=AlGaAs mun=2170 mup=350 tmun=1 tmup=2.1

material region=1 tc.a=2.27 tc.b=0 tc.c=0
material region=2 tc.a=2.27 tc.b=0 tc.c=0
material region=3 tc.a=7.9  tc.b=0 tc.c=0
material region=4 tc.a=2.27 tc.b=0 tc.c=0
material region=5 tc.a=1.e4 tc.b=0 tc.c=0

```

```
138
139 model srh  fldmob  print hcte.el lat.temp
140 model material=GaAs conmob evsatmod=1
141 thermcontact num=1 y.min=0.9 y.max=0.9  ext.temp=300 alpha=1500.
142
143 output val.band con.band e.velocity
144
145 solve init
146
147 method newton maxtrap=6  temp.tol=1.e-4  trap itlim=50 \
148   vsatmod.inc=0.01
149
150 solve vbase=0.005
151 solve vbase=0.02
152 solve vbase=0.05
153 solve vbase=0.2 vstep=0.2 vfinal=1.2 electr=2
154
155 contact num=2 current
156 solve i2=1.e-5
157 solve i2=1.2e-5
158 solve i2=1.5e-5
159
160 log outf=hbtex04_neb.log master
161
162 solve vcollector=0.025
163 solve vcollector=0.05
164 solve vcollector=0.1 vstep=0.05 vfinal=2 electr=3
165
166 save outf=hbtex04_neb.str
167
168 tonyplot -overlay  hbtex04_neb.log hbtex04_eb.log -set hbtex04.set
169 quit
170
171
```

12.1.5 hbtex05.in: SiGe HBT Fabrication and Characterization Simulation

Requires: FLASH/BLAZE

This example demonstrates fabrication and analysis of SiGe HBT. A bipolar device with a SiGe base created in ATHENA, and then passed on to ATLAS for electrical analysis to extract a gummel plot and bipolar gain. It shows:

- SiGe simulation in ATHENA
- specification of donor and acceptor dopants for SiGe material

- ATHENA-ATLAS automatic interface
- DC simulation for Ic and Ib versus Vbe

This example contains two parts. The first part is the ATHENA process simulation of the HBT and the second part is the ATLAS electrical analysis. The whole of the ATLAS part of this example is as described in the first example in this section with the exception of the ATLAS structure specification syntax.

The ATHENA run starts by defining the mesh and the silicon substrate. After the buried collector formation, the structure is flipped upside down using:

```
struct outf=<file>
init inf=<file> flip.y
```

The next step is the deposition of the silicon germanium layer. SiGe is a recognized material in FLASH so no special syntax is needed. The remainder of the bipolar processing is typical process simulation commands for base and emitter implant, silicon emitter layer and various drive steps. Metal contacts are deposited, patterned and defined using the `electrode` statement.

The key statement required for the ATHENA/ATLAS interface is to define the type of dopants that boron and phosphorus are in SiGe. This is necessary for all materials in FLASH. The command,

```
impurity i.boron acceptor sige
```

sets boron as an acceptor-type dopant in SiGe. A similar statement sets phosphorus as a donor-type dopant.

The line `go atlas` defines the automatic interface with ATLAS. In ATLAS, no structure specification syntax (including the mesh statement) is needed.

Results of the electrical simulation can be seen in TONYPLOT. Ic and Ib plots vs. Vbe can be seen from the log file. The band diagram and other physical variables can be seen from the solution file.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

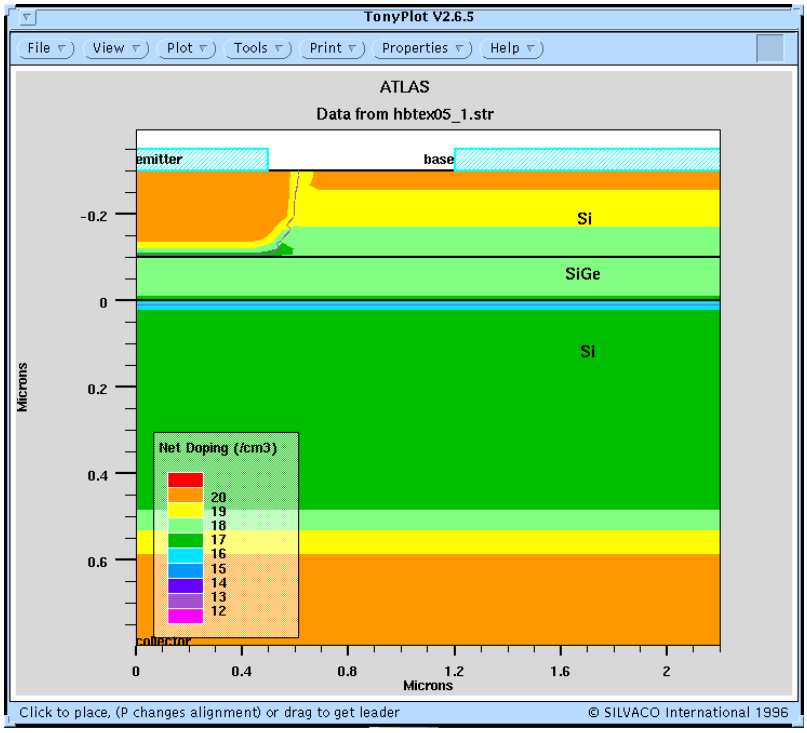


Figure 12.8: Doping, heterojunction and electrode positions for an SiGe HBT defined using FLASH

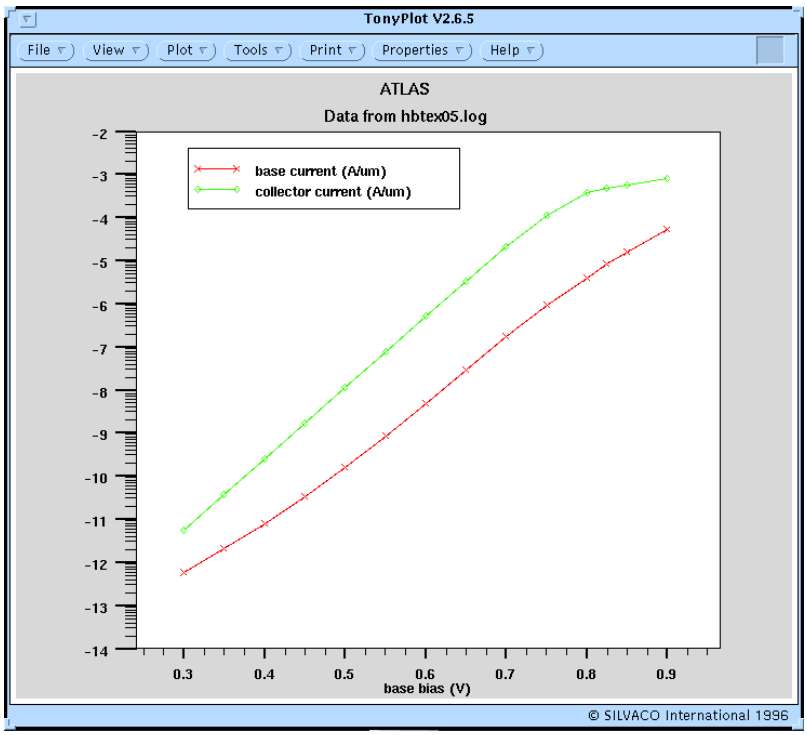


Figure 12.9: Gummel Plot for the SiGe HBT from FLASH

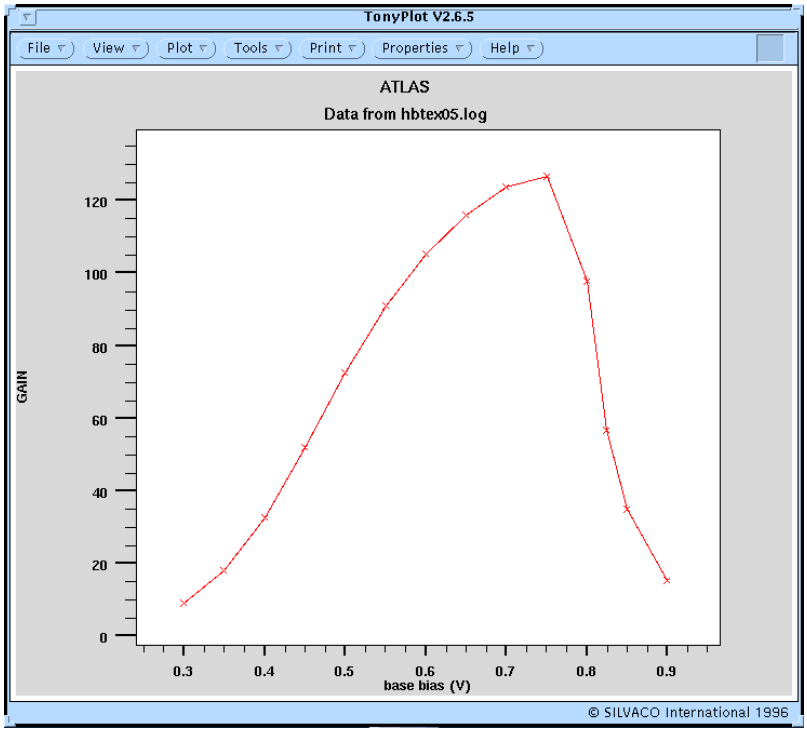


Figure 12.10: Gain versus Vbe for the SiGe HBT

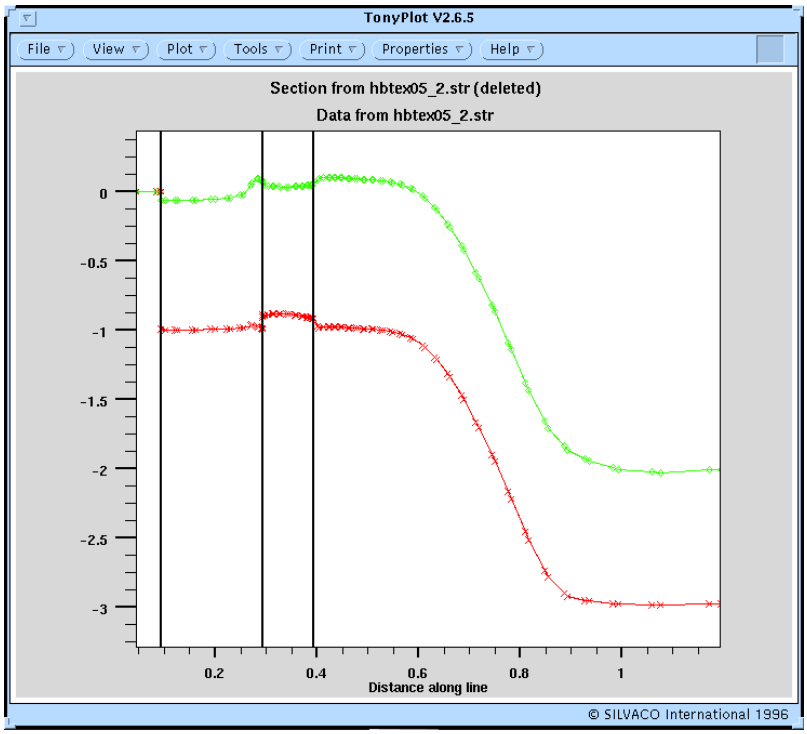


Figure 12.11: Band Diagram of the SiGe HBT with applied VCE

Input File hbt/hbtex05.in:

```
1    go athena
```

```
# SiGe HBT simulation


# Establish initial grid and substrate material

line x  location=0.0   spacing=0.1
line x  location=0.5   spacing=0.05
line x  location=0.7  spacing=0.05
line x  location=1.2  spacing=0.1
line x  location=2.2   spacing=0.25
#
line y  location=0.0   spacing=0.01
line y  location=0.1   spacing=0.02
line y  location=0.5   spacing=0.05
line y  location=0.8   spacing=0.15
#
init silicon c.phos=2e16

structure outf=tmpr1.str

init inf=tmpr1.str flip.y
implant phos energy=60 dose=3e15


diffuse time=5 temp=1000
struct outf=temp
init inf=temp flip.y




# Deposit Silicon germanium with composition fraction 0.2 for base

deposit sige   thick=.1 divis=12 ydy=0.05 dy=0.02 c.frac=0.2 c.boron=1e15

implant   boron  energy=10 dose=1.0e13




# Deposit silicon for the emitter

deposit silicon thick=0.2  divis=10 ydy=0.08 dy=0.04 c.phos=1.e15
```

```
45
46
47
48 implant   boron  energy=12 dose=3e14
49
50
51
52 diffuse time=0.5 temp=920
53
54
55
56 # Mask and implant the emitter
57 deposit photo thick=.5 divis=5
58 etch photo left p1.x=0.5
59 #implant phos energy=40 dose=6e15
60 implant phos energy=38 dose=6e15
61
62 diffuse time=5 temp=920
63 strip
64
65
66
67 # Deposit and pattern the contact metal
68
69 deposit aluminum thick=.05
70
71 etch aluminum start x=0.5 y=10.
72 etch          cont  x=0.5 y=-10.
73 etch          cont  x=1.2 y=-10.
74 etch          done  x=1.2 y=10.
75
76
77 # Define the electrodes
78 electrode    name=emitter x=0.0
79 electrode    name=base    x=2.0
80 electrode    name=collector backside
81
82
83 # Define impurity characteristics in each material
84
85 impurity i.boron acceptor sige
86 impurity i.phos  donor    sige
87
```

```
88
89  structure outfile=hbtex05_0.str
90
91
92  go atlas
93
94
95  # Material parameter and model specification
96
97  material material=Si   taun0=1e-7 taup0=1e-7
98  material material=SiGe  taun0=1.e-8 taup0=1.e-8
99  model    bgn  consrh  auger fldmob conmob
100
101
102 # Initial solution
103
104
105 solve init
106
107 save outf=hbtex05_1.str
108 tonyplot  hbtex05_1.str -set hbtex05_0.set
109
110 # Bias collector to 2 volts
111 output con.band val.band
112 method gummel  newton trap autonr
113 solve prev
114 solve local vcollector=2.0
115
116
117 # Bias base for Gummel plot
118
119 output con.band val.band
120
121 method  newton  trap
122 solve vbase=0.01
123 solve vbase=0.05
124 solve vbase=0.2
125
126 log outfile=hbtex05.log master
127
128 solve vbase=0.3  vstep=0.05 name=base vfinal=0.9
129
130 save outfile=hbtex05_2.str
```

```
131
132 # plot the  logfile
133 tonyplot  hbtex05.log -set hbtex05.set
134
135
136
137 quit
138
```

12.1.6 hbtex06.in: InGaAs/InP HBT DC and High Frequency Characteristics

Requires: DEVEDIT/BLAZE

In this example an HBT structure based on the InGaAs-InP material system is constructed using DEVEDIT. The structure is then passed to ATLAS for electrical testing. The input file consists of the following two main portions:

- construction of the device in DEVEDIT
- simulation of the Gummel plot in ATLAS
- simulation of the AC parameters
- s-parameter extraction

The first stage of the input constructs the HBT geometry, material regions, doping profiles, and electrodes in DEVEDIT. The n-p-n HBT device is based on lattice matched InGaAs-InP material system. It consists of a highly doped InGaAs cap region, followed by another cap region made of InP. The next region, also made of InP, constitutes the emitter proper. As usual in HBTs the emitter (InP) has a wider energy band gap than the base and collector (InGaAs). The base is followed by the n- subcollector and n+ collector regions. The substrate is made of undoped InP.

The structure was created in DEVEDIT by drawing the device regions in interactive mode and specifying 2D doping distribution. In this example each region was uniformly doped. The Ga composition fraction of 0.47 was also specified in DEVEDIT for each of the InGaAs regions. Finally, the mesh was generated automatically by specifying basic mesh constraints and refining it along x- and/or y-directions in the important areas of the device.

The ATLAS simulation begins by reading in the structure from DEVEDIT. DECKBUILD provides an automatic interface between DEVEDIT and ATLAS so that the structure produced by DEVEDIT is transferred to ATLAS without having to indicate the mesh statement.

The first active statements in the ATLAS portion of the input file are material parameters and models definition. In this example the energy band gap, densities of states and other fundamental material parameters for InGaAs, are calculated based on the composition specified in DEVEDIT. For InP the respective default values are used. The band alignment is defined using the `align` parameter on the `material` statement. Material and model parameters can be specified in ATLAS on a material-by-material or a region-by-region basis. The latter possibility is used here to define carrier lifetimes, low field mobilities and saturation velocities, taking into consideration the doping level in the respective regions.

The same set of physical model is applied here to all the regions/materials: Shockley-Read-Hall recombination, electric field dependent mobilities with GaAs-like velocity-field characteristic, and Fermi-Dirac statistics. To reflect the different properties of materials with regard to the critical field `ecritn` is specified in separate `model` statements for InP and InGaAs.

The simulation is first performed to obtain the Gummel plot by biasing simultaneously the base and collector with respect to the emitter up to 1.2 V. At the same time, a small signal AC perturbation

is applied at a frequency of 1 MHz to calculate the AC parameters (conductances and capacitances) as functions of Vbe. This also allows the user to obtain the cutoff frequency by using the low frequency approximation in TONYPLOT or by using the extraction feature of DECKBUILD. Currents, voltages and AC parameters are saved in a log file, and internal structure information is saved for a bias condition (Vbe=Vce=0.95 V) where the cutoff frequency is close to its maximum. The calculation of s-parameters is specified on the log statement with the s.param command.

At the end of simulation, `extract` statements are used to determine the maximum cutoff frequency, the base bias, the input (base) capacitance, and the transconductance - all at the bias of Vbe=Vce=1.0 V where the maximum cutoff frequency was observed.

The structure plot, the Gummel plot, the AC current gain versus frequency, and S-parameters are then displayed using TONYPLOT.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

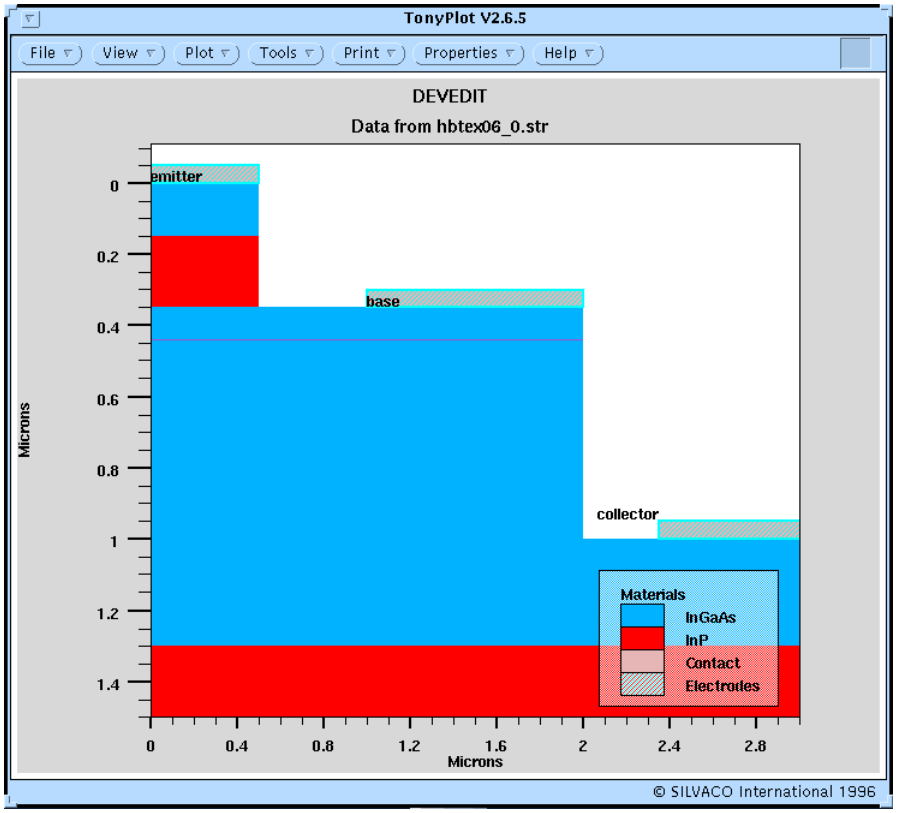


Figure 12.12: InGaAs/InP HBT structure defined in DevEdit

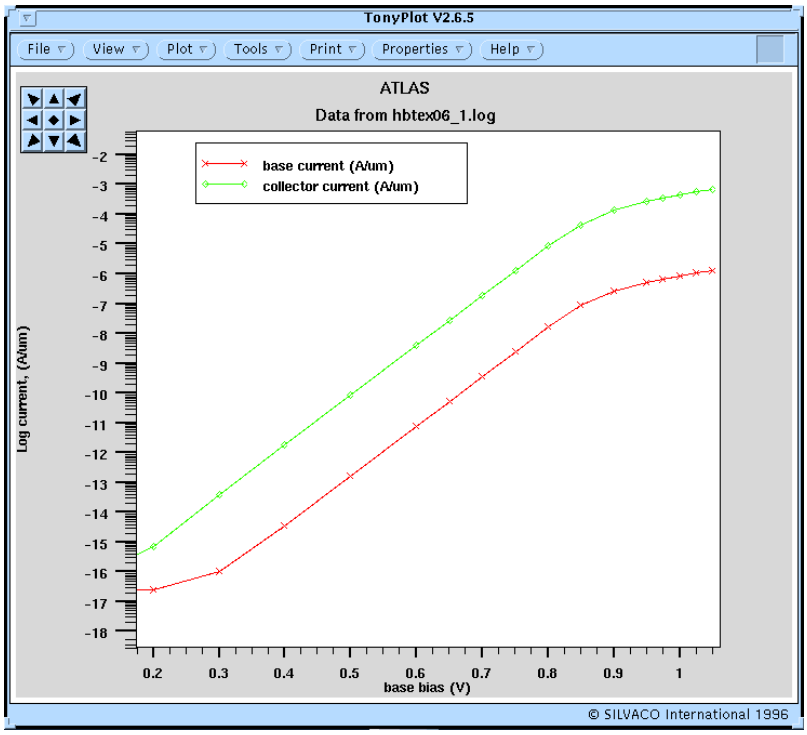


Figure 12.13: Gummel plot for the InGaAs/InP HBT

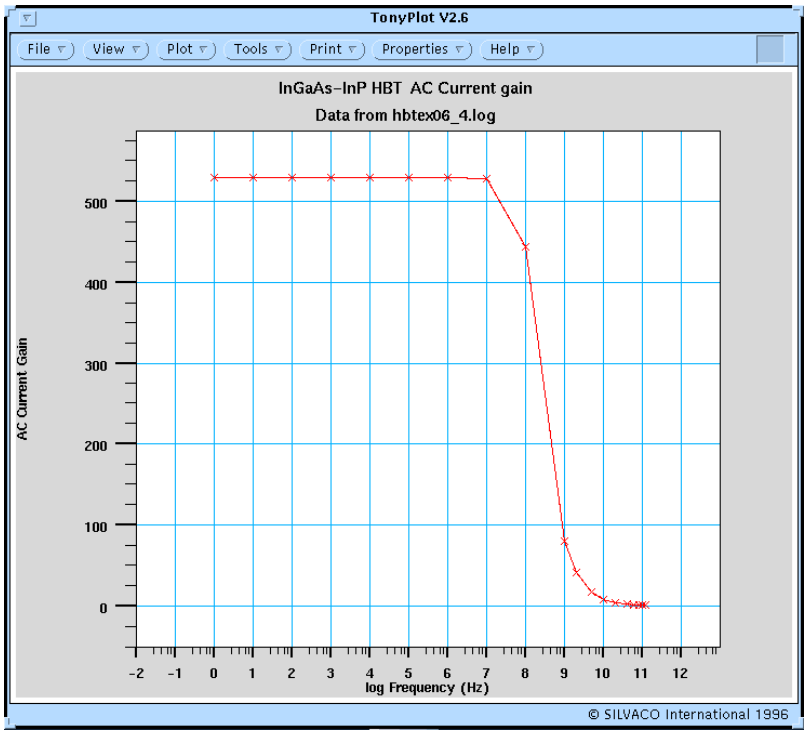


Figure 12.14: fT extraction for InP-based HBT. AC current gain can be extracted at each frequency. fT is where gain is unity

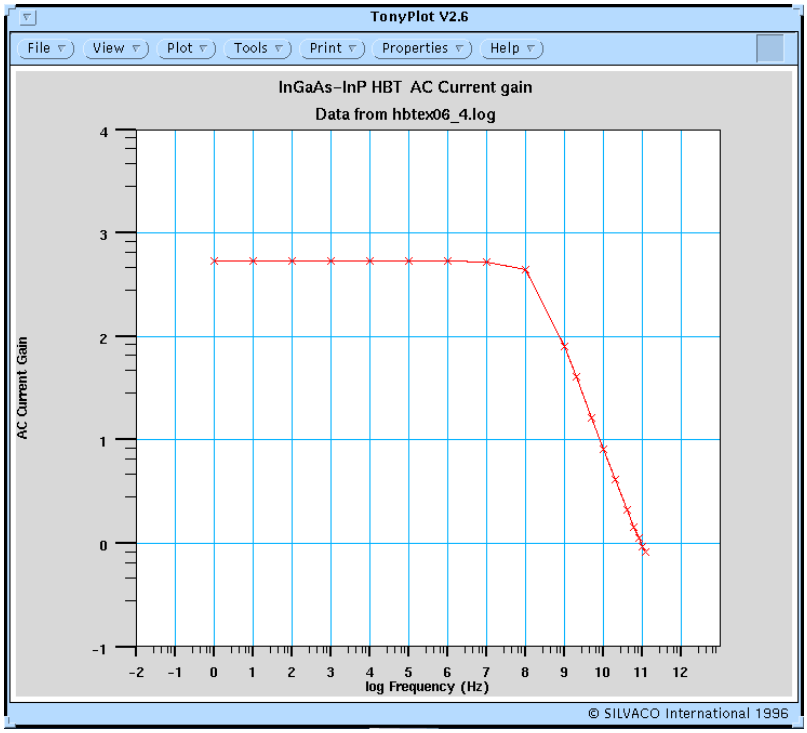


Figure 12.15: fT extraction is often easier on a log scale. Zero on the y-axis is unity gain.

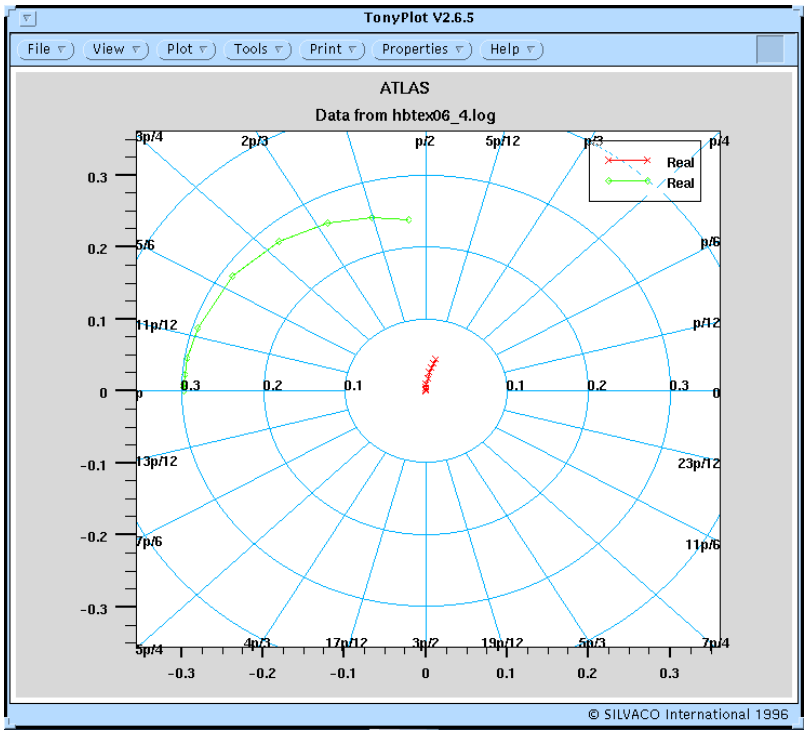


Figure 12.16: Polar plot of InP HBT s-parameters

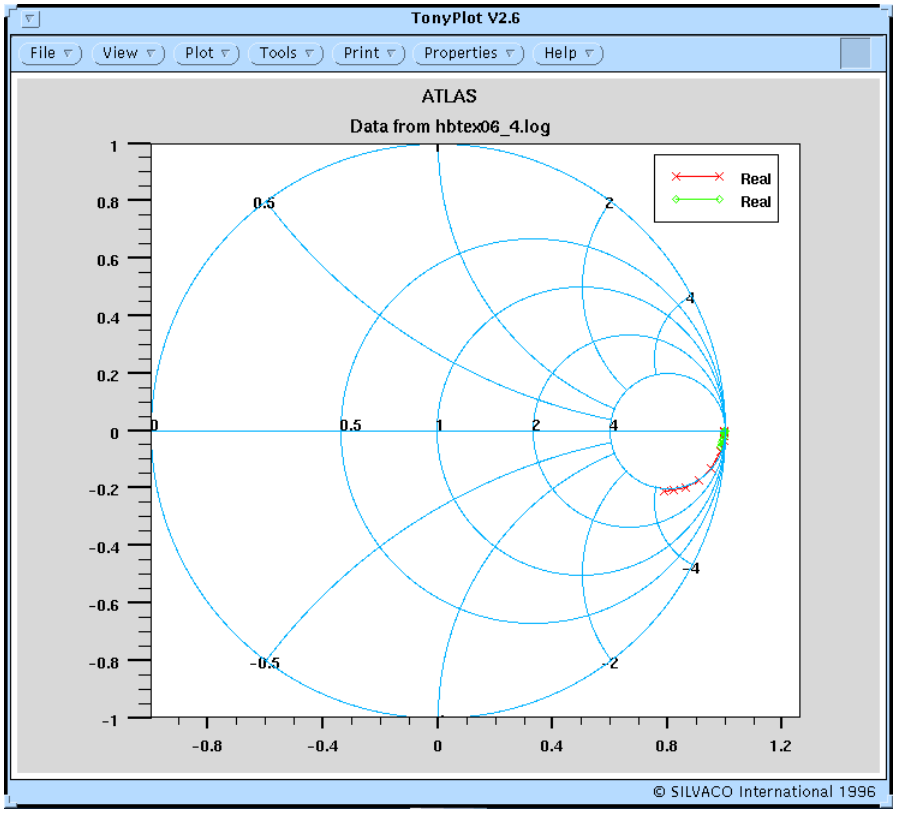


Figure 12.17: Smith Chart of S11 (x) and S22 (o)

Input File hbt/hbtex06.in:

```
go DevEdit

region reg=1 mat=InGaAs \
points="0,0 0.5,0 0.5,0.15 0,0.15 0,0"
#
impurity id=1 region.id=1 imp="Composition Fraction X" \
x1=0 x2=0 y1=0 y2=0 \
peak.value=0.47 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
impurity id=2 region.id=1 imp=Donors \
x1=0 x2=0 y1=0 y2=0 \
peak.value=1e+19 ref.value=1000000000000 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant

region reg=2 mat=InP \
points="0,0.15 0.5,0.15 0.5,0.25 0,0.25 0,0.15"
```

```
#
impurity id=1 region.id=2 imp=Donors \
x1=0 x2=0 y1=0 y2=0 \
peak.value=2e+18 ref.value=1000000000000 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant

region reg=3 mat=InP \
points="0,0.25 0.5,0.25 0.5,0.35 0,0.35 0,0.25"
#
impurity id=1 region.id=3 imp=Donors \
x1=0 x2=0 y1=0 y2=0 \
peak.value=4e+17 ref.value=1000000000000 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant

region reg=4 mat=InGaAs \
points="0,0.35 0.5,0.35 1,0.35 2,0.35 2,0.42 0,0.42 0,0.35"
#
impurity id=1 region.id=4 imp=Acceptors \
x1=0 x2=0 y1=0 y2=0 \
peak.value=2e+19 ref.value=1000000000000 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
impurity id=2 region.id=4 imp="Composition Fraction X" \
x1=0 x2=0 y1=0 y2=0 \
peak.value=0.47 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant

region reg=5 mat=InGaAs \
points="0,0.42 2,0.42 2,1 0,1 0,0.42"
#
impurity id=1 region.id=5 imp="Composition Fraction X" \
x1=0 x2=0 y1=0 y2=0 \
peak.value=0.47 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
impurity id=2 region.id=5 imp=Donors \
x1=0 x2=0 y1=0 y2=0 \
peak.value=1e+16 ref.value=1000000000000 comb.func=Multiply \
```

```
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant

region reg=6 mat=InGaAs \
points="0,1 2,1 2.35,1 3,1 3,1.3 0,1.3 0,1"
#
impurity id=1 region.id=6 imp="Composition Fraction X" \
x1=0 x2=0 y1=0 y2=0 \
peak.value=0.47 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
impurity id=2 region.id=6 imp=Donors \
x1=0 x2=0 y1=0 y2=0 \
peak.value=6e+18 ref.value=1000000000000 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant

region reg=7 mat=InP \
points="0,1.3 3,1.3 3,1.5 0,1.5 0,1.3"
#
impurity id=1 region.id=7 imp=Donors \
x1=0 x2=0 y1=0 y2=0 \
peak.value=10000000000000 ref.value=1000000000000 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant

region reg=8 name=emitter mat=Contact elec.id=1 work.func=0 \
points="0,0 0,-0.05 0.5,-0.05 0.5,0 0,0"

region reg=9 name=base mat=Contact elec.id=2 work.func=0 \
points="1,0.35 1,0.3 2,0.3 2,0.35 1,0.35"

region reg=10 name=collector mat=Contact elec.id=3 work.func=0 \
points="2.35,1 2.35,0.95 3,0.95 3,1 2.35,1"



# Set Meshing Parameters
#
base.mesh height=0.1 width=0.125
#
```

```
105 bound.cond max.slope=28 max.ratio=300 rnd.unit=0.001 line.straightening=1
      \
106     align.points when=automatic
107 #
108 constr.mesh max.angle=90 max.ratio=200 max.height=1 \
109     max.width=1 min.height=0.0001 min.width=0.0001
110 #
111 constr.mesh  x1=0 x2=0.5  y1=-0.05 y2=0.325 max.height=0.04
112 constr.mesh  x1=0 x2=0.5  y1=0.325 y2=0.35  max.height=0.02
113 constr.mesh  x1=0 x2=2.0  y1=0.35  y2=0.5   max.height=0.02
114 constr.mesh  x1=0 x2=2.0  y1=0.5   y2=0.7   max.height=0.04
115 constr.mesh  x1=0 x2=2.0  y1=0.95  y2=1.0   max.height=0.04
116 constr.mesh  x1=0 x2=3.0  y1=1.0   y2=1.05  max.height=0.04
117 #
118 # Perform mesh operations
119 #
120 mesh
121 #
122 structure outf=hbtex06_0.str
123 tonyplot  hbtex06_0.str -set hbtex06_0.set
124
125 go atlas
126 #
127 title  InGaAs/InP HBT simulation
128 #
129 # Material and model specification
130 #
131 material material=InGaAs align=0.36
132 material material=InP    align=0.36
133 #
134 # Cap1         InGaAs region
135 material region=1 taun0=5.0e-10 taup0=1.0e-9 vsatn=2.5e7 mun0=4000
      mup0=200
136 #
137 # Cap2         InP    region
138 material region=2 taun0=5.0e-9  taup0=5.0e-9 vsatn=2.0e7 mun0=2000
      mup0=100
139 #
140 # Emitter      InP    region
141 material region=3 taun0=1.0e-8  taup0=2.0e-8 vsatn=2.0e7 mun0=3000
      mup0=160
142 #
143 # Base         InGaAs region p-type
```

```
material region=4 taun0=3.0e-10 taup0=3.0e-10 vsatn=2.5e7 mun0=5000
  mup0=240
#
# Collector    InGaAs region
material region=5 taun0=1.0e-7  taup0=1.0e-7 vsatn=2.5e7  mun0=10000
  mup0=400
#
# Subcollector InGaAs region
material region=6 taun0=1.0e-9 taup0=1.0e-9 vsatn=2.5e7   mun0=4000
  mup0=200
#
model material=InP    srh optr fldmob evsatmod=1 ecritn=6.e3 fermidirac
  print
model material=InGaAs srh optr fldmob evsatmod=1 ecritn=3.e3 fermidirac
  print
#
# Initial solution
output con.band val.band
solve init
#
# Calculate Gummel plot and AC parameters versus Vbe (Vce) at 1 MHz
solve prev

#
# 1 - emitter  2 - base  3 - collector
log outf=hbtex06_1.log

solve v2=0.01  v3=0.01  ac freq=10
solve v2=0.025 v3=0.025 vstep=0.025 electr=23 nstep=2 ac freq=1e6
solve v2=0.1   v3=0.1   vstep=0.1   electr=23 nstep=5 ac freq=1e6
solve v2=0.65  v3=0.65  vstep=0.05  electr=23 nstep=6 ac freq=1e6

save  outf=hbtex06_2.str

solve v2=0.975 v3=0.975 vstep=0.025 electr=23 nstep=3 ac freq=1e6
#
# Frequency domain AC analysis up to 100 GHz
#

log outf=hbtex06_3.log s.param inport=base outport=collector


load inf=hbtex06_2.str master.in

```

```
183 solve previous ac freq=1
184 solve  ac freq=10 fstep=10 mult.f nfstep=8
185 solve  ac freq=2e9
186 solve  ac freq=5e9
187 solve  ac freq=1e10
188 solve  ac freq=2e10 fstep=2e10 nfstep=5
189 #
190 #  Extraction of parameters
191 #
192
193 extract init inf="hbtex06_1.log"
194 #
195 # Maximum cutoff frequency
196 extract name="Ft_max" max(g."collector""base"/(6.28*c."base""base"))
197 #
198 # Base bias at maximum cutoff frequency
199 extract name="Vbe@Ft_max" x.val from curve (v."base", g."collec-
      tor""base"/(6.28*c."base""base" )) where y.val=$"Ft_max"
200 #
201 # Input (base) capacitance at maximum cutoff frequency
202 extract name="Cbb@Ft_max)" y.val from curve (v."base", abs(c."base""base"
      )) where x.val=$"Vbe@Ft_max"
203 #
204 # Transconductance at maximum cutoff frequency
205 extract name="Gm@Ft_max)" y.val from curve (v."base", abs(g."collec-
      tor""base" )) where x.val=$"Vbe@Ft_max"
206 #
207 #
208 # Gummel plot
209
210 tonyplot  hbtex06_1.log -set hbtex06_1_log.set
211
212 #
213 # AC current gain versus frequency
214 tonyplot  hbtex06_4.log -set hbtex06_4_log.set
215 #
216 # S12 & S21 polar coordinates
217 tonyplot hbtex06_4.log -set hbtex06_4_s12.set
218 #
219 # S11& S22 Smith chart
220 tonyplot hbtex06_4.log -set hbtex06_4_s11.set
221 #
222 quit
223
```

This page is intentionally left blank

13.1. HEMT: HEMT Application Examples

13.1.1. hemtex01.in: User Model Development with the C-Interpreter

Requires: BLAZE/C-INTERPRETER

This example demonstrates the use of the C-Interpreter to develop user defined material parameters and models for AlGaAs/GaAs HEMT simulation. Here the interpreter is used to prototype a composition dependent electron mobility model for AlGaAs. The interpreted function in this case corresponds to the built-in model for AlGaAs. The two can be compared. The example consists of the following parts:

- construction of the device and the grid using ATLAS syntax
- specification of material regions including a graded AlGaAs heterojunction
- definition of electrodes, doping, material and models parameters
- use of a `C-Interpreter` function for composition and doping dependent electron mobility
- simulation of Id-Vds characteristic
- display of the results in TonyPlot

In the first part of the input file, the device is described using ATLAS syntax including mesh, regions and electrodes locations, and doping distribution. The `region` statements are used to define AlGaAs and GaAs regions. The Al composition fraction (x.composition=0.3) and grading distance are defined here as well.

After the device description, the `material` statement is used to specify some of the material parameters that are different from the default values. These are electron and hole SRH lifetimes (capture times) which apply to both GaAs and AlGaAs material regions. In the same statement we indicate that the low field electron mobility as a function of composition and doping will be calculated using the `C-Interpreter`. The name of the function and the name of the corresponding file in which this function is described are specified as a parameter,

```
f.conmun=hemtex01_interp.lib
```

in the `material` statement. Additionally, the `alignment` parameter is specified which defines the band offset between the conduction and valance bands. The value of 0.6 sets 60% of the step to the conduction band.

The `models` statement is used to specify the following set of models: band-gap narrowing, field dependent mobility and the Shockley-Read-Hall recombination. The two-carrier transport model is specified here by the parameter `carriers=2`. The `contact` statement is used to set the workfunction at the Schottky gate electrode.

The solution procedure begins from the initial solution at zero bias or thermodynamic equilibrium. The structure of the device at zero bias is displayed using TONYPLOT. For subsequent simulation, the combined Gummel-Newton iteration method is specified in the `method` statement. The `output` statement is used to include additional parameters in the output structure file: conduction and valence band potentials, and electron and hole mobilities. The gate bias is then ramped to -0.6V. For this gate bias, the Id-Vd characteristic is calculated by sweeping the drain voltage up to 5V. The results of simulation are saved in the log file and then displayed using TONYPLOT.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into the DECKBUILD, select the **run** button to execute the example.

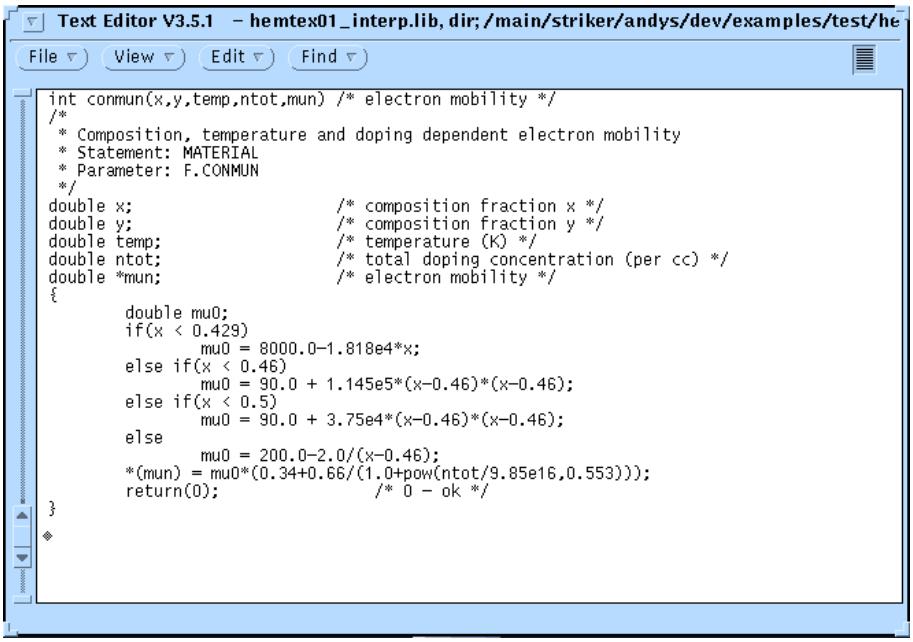


Figure 13.1: User defined C-Interpreter Function called by ATLAS

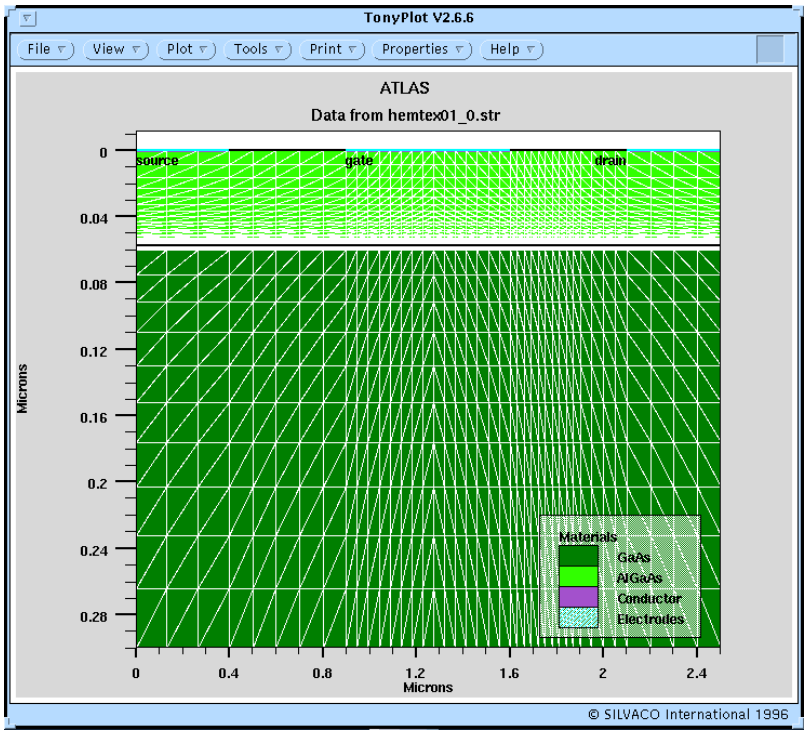


Figure 13.2: HEMT material structure and mesh defined using ATLAS syntax

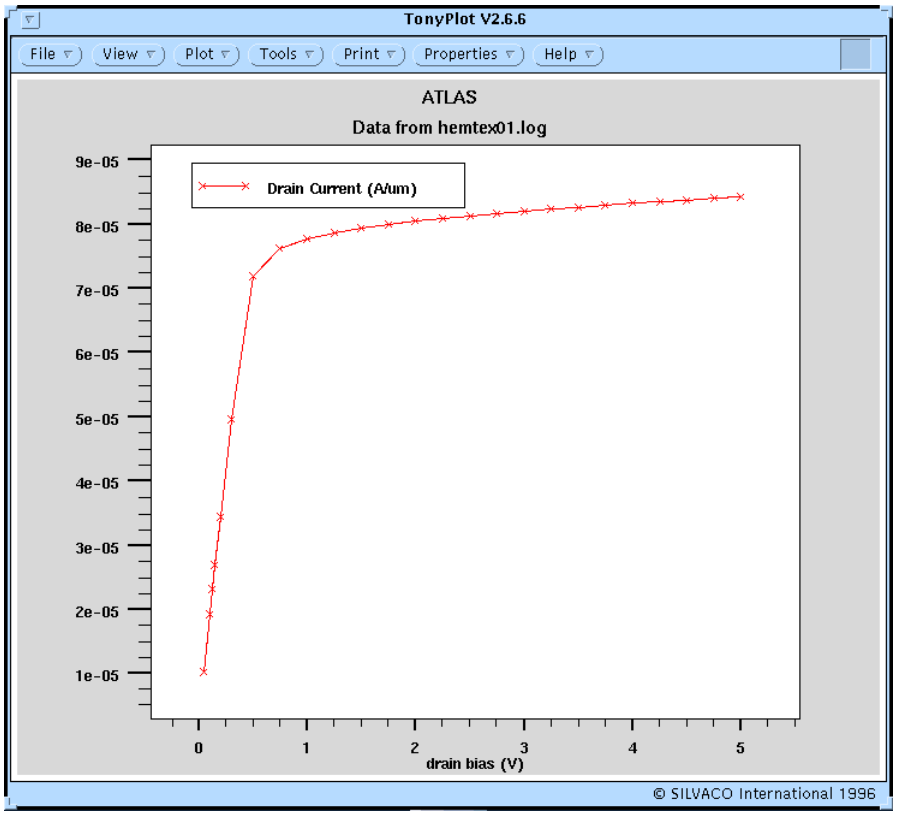


Figure 13.3: IdVd curve using the user-defined interpreter model

Input File hemt/hemtex01.in :

```
1   go atlas
2   #
3   # SILVACO International 1996
4   #
5   # SECTION 1: Mesh input
6   #
7   mesh       nx=42 ny=38
8   x.m        n=1    l=0.0   r=1.0
9   x.m        n=4    l=0.4   r=1.0
10  x.m        n=9    l=0.9   r=1.0
11  x.m        n=24   l=1.6   r=1.0
12  x.m        n=34   l=1.9   r=1.0
13  x.m        n=38   l=2.1   r=1.0
14  x.m        n=42   l=2.5   r=1.0
15  #
16  y.m        n=1     l=0.0         r=1.0
17  y.m        n=15    l=0.0530      r=0.85
18  y.m        n=28    l=0.0600      r=1.0
19  y.m        n=38    l=0.3000      r=1.1
```

```
#
# SECTION 2: Structure Specification
#
region     num=1   material=GaAs y.min=0.055
region     num=2  material=AlGaAs y.max=0.055 x.composition=0.3
  grad.3=0.002
#
elec       num=1  name=source x.min=0.0 x.max=0.4 y.min=0.0 y.max=0.0
elec       num=2  name=gate   x.min=0.9 x.max=1.6 y.min=0.0 y.max=0.0
elec       num=3  name=drain  x.min=2.1 x.max=2.5 y.min=0.0 y.max=0.0
#
doping     uniform y.max=0.0530 n.type conc=1.e18
doping     uniform y.min=0.0530 n.type conc=1.e14
#
# SECTION 3: Material Models
#
material taun0=1.e-9 taup0=1.e-9 f.conmun=hemtex01_interp.lib
material align=0.6
#
model    bgn fldmob srh
#
contact name=gate workfun=4.87
#
# SECTION 4: Initial solution
#
solve init
save outf=hemtex01_0.str
tonyplot  hemtex01_0.str -set hemtex01_0.set
#
# SECTION 3: Bias gate
#
method gummel  newton trap itlimit=20 maxtrap=6
output con.band val.band e.mobility h.mobility
solve vgate=-0.0 vstep=-0.2 name=gate vfinal=-0.6
#
# SECTION 6: Drain ramp
#
log outf=hemtex01.log master
solve vdrain=0.05
solve vdrain=0.10
solve vdrain=0.125
solve vdrain=0.15
solve vdrain=0.20
```

```
62 solve vdrain=0.30
63 #
64 method  newton trap itlim=20 maxtrap=6
65 solve vdrain=0.50  vstep=0.25 name=drain vfinal=5.0
66 tonyplot hemtex01.log -set hemtex01.set
67 #
68 quit
```

13.1.2. hemtex02.in: Lattice Matched HEMT Breakdown Simulation

Requires: BLAZE

This example demonstrates simulation of avalanche breakdown in a single quantum-well lattice matched AlGaAs/GaAs HEMT. It shows:

- construction of the heterojunction structure using ATLAS syntax
- material and models parameter specification
- simulation of Id-Vds and breakdown characteristic
- display of the results in TONYPLOT

The device under consideration consists of the highly doped (1.e18/cm3) AlGaAs layer with the Al composition fraction of 0.3, the GaAs channel layer 250 A thick with the donor concentration of 1.e15 /cm3, and AlGaAs buffer layer with the same doping. The parts of the channel under the source and drain are heavily doped, and source and drain electrodes are located vertically, touching the channel region. This technique accounts for the alloying of contact material and the tunneling of carriers through the heterojunction. The channel length is 0.5 micron.

In the first part of the input file, the device is described using using the ATLAS structural syntax. The description includes the mesh, regions locations, electrodes locations, and doping distribution. The `region` statements are used to define the AlGaAs and GaAs regions. The Al composition fraction (x.composition=0.3) is defined here as well. Note that an artificial oxide layer (region num=4) is defined here providing the possibility to specify AlGaAs surface states using the `interface` statement.

After the device description, the first `material` statement is used to specify the electron and hole SRH lifetimes applied to both GaAs and AlGaAs materials. In the second `material` statement the low field mobilities, electron affinity, and density of states in the conduction band Nc are defined for AlGaAs. In this example the band alignment is calculated by using energy band gaps and electron affinities of the materials forming heterojunctions. If not explicitly specified in the `material` statements the default values of other material parameters are applied.

The `models` statements are used to specify the following set of models: field dependent mobility, and SRH. The two-carrier transport model is specified here as well (the default). In addition, the concentration dependent mobility model is activated.

In order to simulate avalanche breakdown, the impact ionization-generation model should be turned on. This is done using the `impact selb` statement in which the Selberherr impact ionization model is activated. All basic impact ionization parameters are user-accessible and can be modified by the user. As an example, the appropriate set of parameters is defined in this statement. The parameters apply to both GaAs and AlGaAs materials.

The gate electrode in HEMT structures is of the Schottky type. To indicate this, the `contact` statement is used to define the workfunction of the gate electrode, which gives the Schottky barrier height of approximately 0.9V.

The initial solution is performed automatically if not specified. The gate voltage is set to zero, the structure under zero bias is displayed using TonyPlot, and the drain voltage is ramped up to 0.3V. The solutions are obtained using the combined Gummel-Newton algorithm specified in the statement, `method gummel newton`. The algorithm implies that if the solution does not converge in the course of the decoupled Gummel iterations, the program will automatically switch over to the fully coupled Newton algorithm.

Next, the drain voltage is ramped until the drain current reaches the predefined value of 9.e-5 A/um, well within the breakdown region. Using the **compliance limits on the solve statement.** These calculations are performed using the **Newton** method :

```
method newton
```

The results of simulation are saved in the log file and the Id-Vds breakdown characteristics are displayed using TONYPLOT.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into the DECKBUILD, select the **run** button to execute the example.

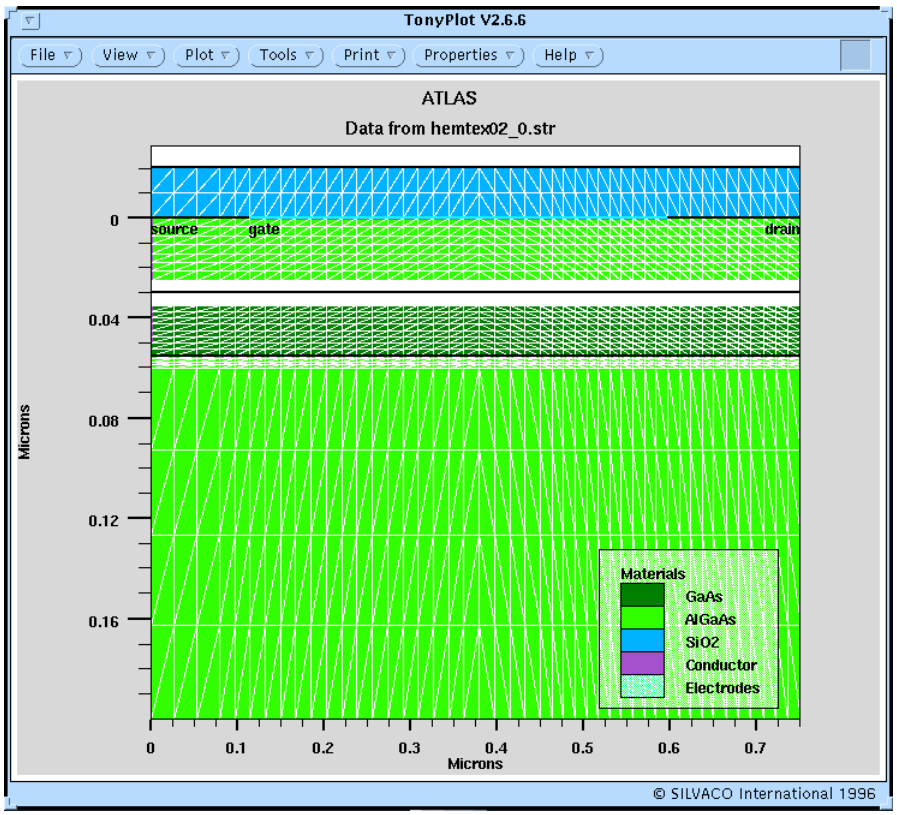


Figure 13.4: HEMT Structure and Mesh. Note the fine mesh recommended at the heterojunctions

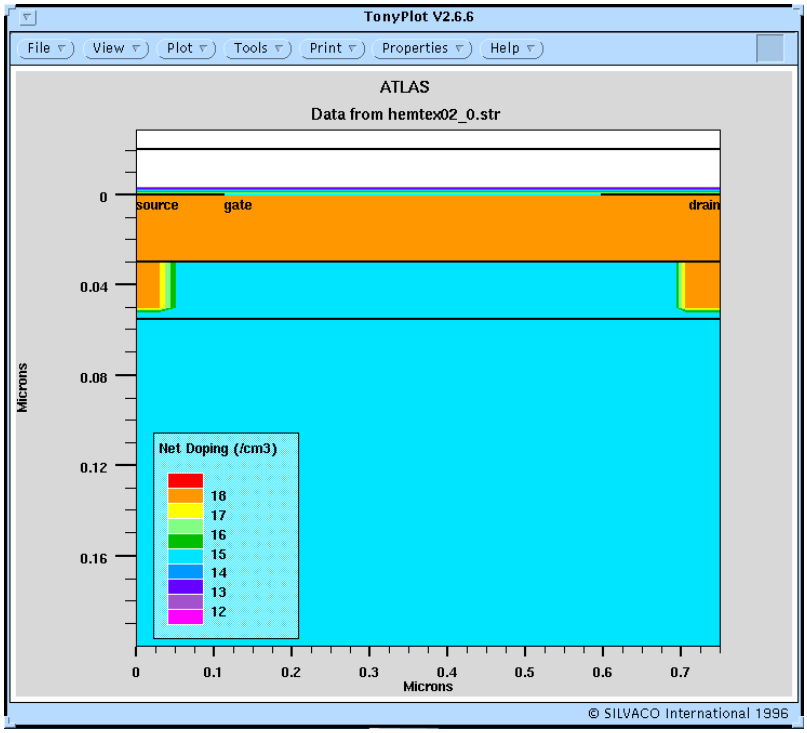


Figure 13.5: HEMT doping profile

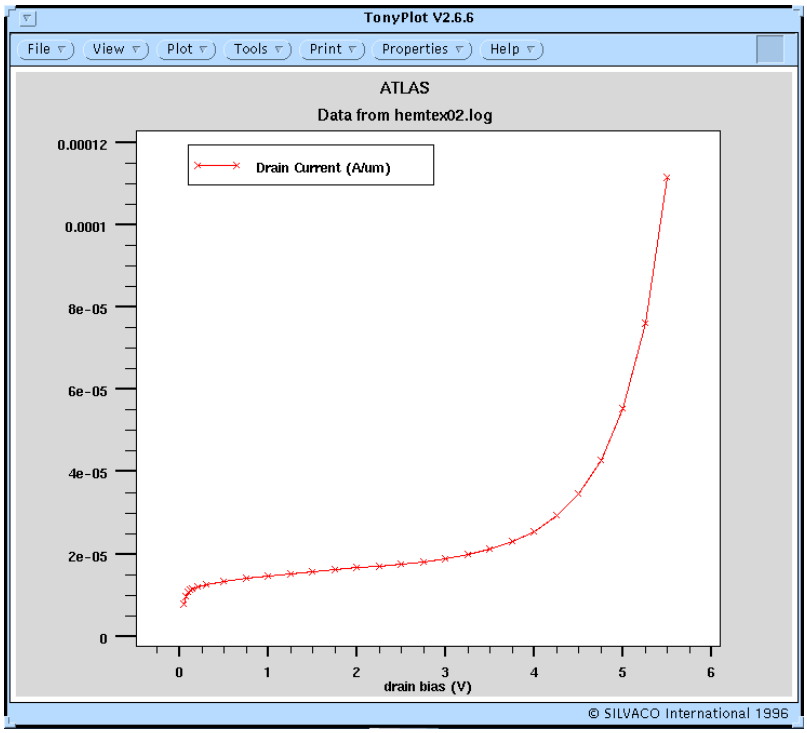


Figure 13.6: HEMT Id/Vds curve including avalanche portion

Input File hemt/hemtex02.in :

```
1   go atlas
2   Title    HEMT breakdown characteristics
3   #
4   # SILVACO International 1996
5   #
6   # SECTION 1: Mesh input
7   #
8   mesh         nx=47 ny=47
9   #
10  x.m        n=1    l=0.0   r=1.0
11  x.m        n=4    l=0.08   r=1.0
12  x.m        n=8    l=0.15   r=1.0
13  x.m        n=25   l=0.45   r=1.0
14  x.m        n=40   l=0.65   r=1.0
15  x.m        n=47   l=0.75   r=1.0
16  #
17  y.m        n=1     l=-0.02   r=1.0
18  y.m        n=3     l=0.0     r=1.0
19  y.m        n=10    l=0.025   r=1.0
20  y.m        n=20    l=0.03    r=1.0
21  y.m        n=31    l=0.035   r=1.0
22  y.m        n=39    l=0.055   r=1.0
23  y.m        n=43    l=0.06    r=1.0
24  y.m        n=47    l=0.20    r=1.05
25  #
26  #
27  # SECTION 2: Structure Specification
28  #
29  region     num=1   material=GaAs y.min=0.03 y.max=0.055
30  region     num=2  material=AlGaAs y.max=0.03 x.composition=0.3
31  region     num=3  material=AlGaAs y.min=0.055 x.composition=0.3
32  region     num=4  oxide y.min=-0.02 y.max=0
33  #
34  elec       num=1  name=source x.min=0.0 x.max=0.0 y.min=0.0 y.max=0.05
35  elec       num=2  name=gate   x.min=0.1 x.max=0.6 y.min=0.0 y.max=0.0
36  elec       num=3  name=drain  x.min=0.75 x.max=0.75 y.min=0.0 y.max=0.05
37  #
38  doping     uniform y.min=0 y.max=0.03 n.type conc=1.e18
39  doping     uniform y.min=0.03 n.type conc=1.e15
40  doping     uniform x.min=0.0 x.max=0.05 y.min=0.03 y.max=0.05 \
41     n.type conc=1.e18
42  doping     uniform x.min=0.70 x.max=0.75 y.min=0.03 y.max=0.05 \
```

```
43     n.type conc=1.e18
44  #
45  interface x.min=0 x.max=0.75 y.min=-0.01 y.max=0.005 qf=-1.e12
46  #
47  #
48  # SECTION 3: Material Models
49  #
50  material taun0=1.e-9 taup0=1.e-9
51  material material=AlGaAs mun=2000 mup=350 affinity=3.82  nc300=5.7e17
52  material align=0.6
53  #
54  model    fldmob srh
55  model    material=GaAs conmob
56  #
57  impact  selb an1=1.899e5 bn1=5.75e5 ap1=2.215e5 bp1=6.57e5 \
58  an2=1.899e5 bn2=5.75e5 ap2=2.215e5 bp2=6.57e5 egran=1.e20 betan=1.82 be-
      tap=1.75
59  #
60  contact  name=gate workfun=4.73
61  #
62  # SECTION 4: Bias gate
63  #
64  method gummel  newton itlim=20 trap  maxtrap=6
65  output con.band val.band
66  #
67  solve vgate=0
68  save outf=hemtex02_0.str
69  tonyplot hemtex02_0.str -set hemtex02_0.set
70  #
71  # SECTION 5: Drain ramp
72  #
73  log outf=hemtex02.log master
74  solve vdrain=0.05
75  solve vdrain=0.10
76  solve vdrain=0.125
77  solve vdrain=0.15
78  solve vdrain=0.20
79  solve vdrain=0.30
80  #
81  method newton trap itlim=35  maxtrap=6
82  solve vdrain=0.50  vstep=0.25  name=drain vfinal=6 compl=9.e-5 e.comp=3
83  #
84  tonyplot  hemtex02.log -set hemtex02.set
```

```
85 #
86 quit
```

13.1.3. hemtex03.in: AlGaAs/GaAs HEMT Id-Vgs and Id-Vds Characterization

Requires: BLAZE

This example demonstrates Id-Vds and Id-Vgs calculations in a single quantum-well AlGaAs/GaAs HEMT. It shows:

- construction of the heterojunction structure using ATLAS syntax
- material and models parameter specification
- simulation of Id/Vds and Id/Vgs characteristics
- display of the results in TONYPLOT

The device under consideration is the same one described as in the previous example in this section. The devices geometry, mesh, regions, electrodes and doping distribution are described with the same sequence of statements as in the example above. The same material parameters and physical models are applied, except for the impact ionization-generation model. Since the breakdown analysis is not the goal of this example and the range of voltages applied are far away from the breakdown region where the impact ionization effects become essential, the `impact` statement is excluded and no impact ionization-generation is considered. The contact and interface statements are also a repeat of those from the previous example.

After the initial solution is obtained, the gate voltage is set to 0, the structure under zero bias is displayed using TonyPlot, and the Id-Vds characteristic is calculated. As in the previous example, the drain voltage is first ramped up to 0.3V. The solutions are obtained using the combined Gummel-Newton algorithm specified in the `method gummel newton` statement. The algorithm implies that if the solution does not converge in the course of Gummel iterations, the program will automatically switch over to the Newton algorithm.

Next, the drain voltage is ramped up to 3V. This part of the simulation is performed using the Newton method: `method newton`. The simulation results are saved in the log file and displayed using TONYPLOT.

Simulation of the Id-Vgs characteristic is performed next. At the first stage the solution for Vgs=-0.9V and Vds=0.5V is obtained. Next, the gate voltage is ramped up to 0.6V with the step of 0.1V.

The simulation results are again saved in a separate log file and displayed using TONYPLOT.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into the DeckBuild, select the **run** button to execute the example.

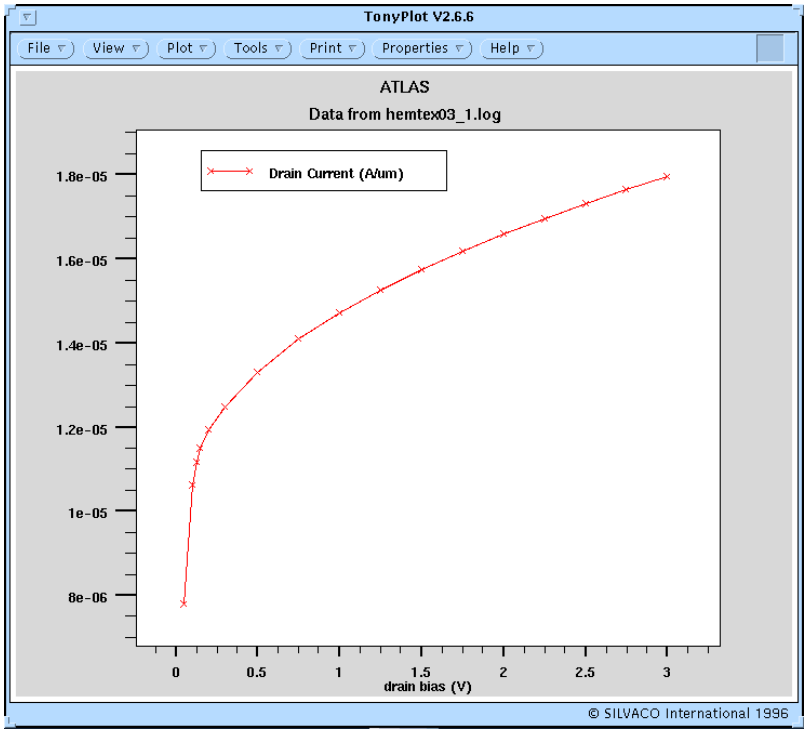


Figure 13.7: Single well HEMT Id/Vds curve

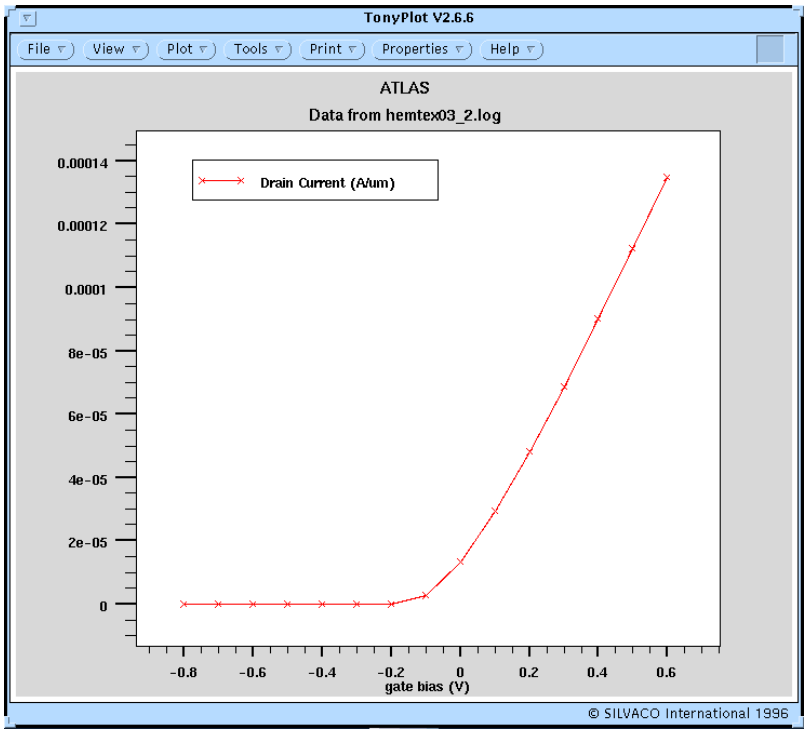


Figure 13.8: Single well HEMT Id/Vgs curve to extract threshold voltage

Input File hemt/hemtex03.in:

```
1    go atlas
```

```
Title    HEMT Id-Vd and Id-Vg characteristics
#
# SILVACO International 1996
#
# SECTION 1: Mesh input
#
mesh       nx=47 ny=47
#
x.m        n=1    l=0.0   r=1.0
x.m        n=4    l=0.08   r=1.0
x.m        n=8    l=0.15   r=1.0
x.m        n=25   l=0.45   r=1.0
x.m        n=40   l=0.65   r=1.0
x.m        n=47   l=0.75   r=1.0
#
y.m        n=1     l=-0.02   r=1.0
y.m        n=3     l=0.0     r=1.0
y.m        n=10    l=0.025   r=1.0
y.m        n=20    l=0.03    r=1.0
y.m        n=31    l=0.035   r=1.0
y.m        n=39    l=0.055   r=1.0
y.m        n=43    l=0.06    r=1.0
y.m        n=47    l=0.20    r=1.05
#
# SECTION 2: Structure Specification
#
region     num=1   material=GaAs y.min=0.03 y.max=0.055
region     num=2  material=AlGaAs y.max=0.03 x.composition=0.3
region     num=3  material=AlGaAs y.min=0.055 x.composition=0.3
region     num=4  oxide y.min=-0.02 y.max=0
#
elec       num=1  name=source x.min=0.0 x.max=0.0 y.min=0.0 y.max=0.05
elec       num=2  name=gate   x.min=0.1 x.max=0.6 y.min=0.0 y.max=0.0
elec       num=3  name=drain  x.min=0.75 x.max=0.75 y.min=0.0 y.max=0.05
#
doping     uniform y.min=0 y.max=0.03 n.type conc=1.e18
doping     uniform y.min=0.03 n.type conc=1.e15
doping     uniform x.min=0.0 x.max=0.05 y.min=0.03 y.max=0.05 \
   n.type conc=1.e18
doping     uniform x.min=0.70 x.max=0.75 y.min=0.03 y.max=0.05 \
   n.type conc=1.e18
#
interface x.min=0 x.max=0.75 y.min=-0.01 y.max=0.005 qf=-1.e12
```

```
45 #
46 # SECTION 3: Material Models
47 #
48 material taun0=1.e-9 taup0=1.e-9
49 material material=AlGaAs mun=2000 mup=350 affinity=3.82  nc300=5.7e17
50 #
51 model    fldmob srh
52 model material=GaAs conmob
53 #
54 contact  name=gate workfun=4.73
55 #
56 # SECTION 4: Id-Vd calculation
57 #
58 method gummel  newton itlim=20 trap  maxtrap=6 vsatmod.inc=0.01 carriers=1
     elect
59 output con.band val.band
60 solve vgate=0
61 #
62 save outf=hemtex03_0.str
63 tonyplot  hemtex03_0.str -set hemtex03_0.set
64 #
65 log outf=hemtex03_1.log  master
66 solve vdrain=0.05
67 solve vdrain=0.10
68 solve vdrain=0.125
69 solve vdrain=0.15
70 solve vdrain=0.20
71 solve vdrain=0.30
72 #
73 method  newton trap itlim=35  maxtrap=6    carriers=1 elect
74 solve vdrain=0.50  vstep=0.25  name=drain vfinal=3
75 #
76 save outf=hemtex03_1.str
77 tonyplot  hemtex03_1.log -set  hemtex03_1log.set
78 #
79 # SECTION 5: Id-Vg calculation
80 #
81 log off
82 solve init
83 method gummel  newton trap itlim=35  maxtrap=6 vsatmod.inc=0.01 carriers=1
     elect
84 solve vgate=-0.9 vdrain=0
85 solve vdrain=0.1  vstep=0.1  name=drain vfinal=0.5
86 #
```

```
log outf=hemtex03_2.log master
method  newton trap itlim=35  maxtrap=6  carriers=1 elect
solve vgate=-0.8  vstep=0.1  name=gate vfinal=0.6
#
save outf=hemtex03_2.str
tonyplot  hemtex03_2.log -set  hemtex03_2log.set
#
quit
```

13.1.4. hemtex04.in: Energy Balance and Drift Diffusion Comparison

Requires: BLAZE

This example demonstrates Id-Vds calculations in single quantum-well AlGaAs/GaAs HEMT using Energy Balance (EB) and Drift Diffusion (DD) Models. It includes

- construction of a heterojunction structure using ATLAS syntax
- material parameter specification
- physical model specification including the energy balance model
- simulation of Id-Vds characteristic with Vgs=0.0V
- repeat simulation using drift-diffusion models for comparison
- display of combined results in TONYPLOT

The example file consists of two ATLAS runs. Both use the same device structure defined in a similar manner to the previous two examples. The first run uses energy balance models to simulate Id-Vds characteristics and the second does the same electrical simulation using the classical drift-diffusion model. The aim of the example is to compare the effects of the different carrier transport models.

Energy balance models provide a more accurate description of physical device effects, in particular the effect of velocity overshoot and non-local impact ionization. These are not handled by the classical drift-diffusion model. In sub-micron HEMT simulation, these two effects can be observed. This example concentrates on the velocity overshoot effect in the GaAs-AlGaAs structure. Non-local impact ionization effects can be seen in breakdown simulations. Since velocity overshoot is not accounted for in the drift-diffusion model the results from simulations with this model will under estimate the current. This discrepancy gets worse as the channel length decreases.

The sequence of simulation syntax is similar to the previous example. For energy balance simulation, additional input parameters need to be defined in, carrier energy and mobility relaxation times in particular. In these examples the electron relaxation times *taurel.el* and *taumob.el* are defined in two separate `material` statements for the GaAs and AlGaAs materials, overriding the default values. In the same statements, the low field electron mobilities are specified for GaAs and AlGaAs. The saturation velocity is also explicitly defined here for GaAs. The `material` statement for AlGaAs also contains the `align` parameter, which defines the portion of the energy band gap difference going to the conduction band (60% in this example) at the GaAs-AlGaAs heterojunction. If the `align` parameter is present in the `material` statement, the energy band alignment is calculated based on the value of this parameter rather than using electron affinities of the materials forming the respective heterojunction.

Since HEMTs are majority carrier devices and the impact-ionization and breakdown are not a subject of analysis here, the one carrier (electrons only) transport model is used in both continuity and energy balance equations. Thus the following three basic equations are solved in this example to simulate the device characteristics: Poisson, electron continuity and energy balance equations.

In the models statements the set of physical models used is specified: field dependent mobility and the one carrier transport model (carriers=1 electron). The solution for the energy balance equation for electrons is activated in this statement using the parameter hcte.el.

The contact statement is used to specify the workfunction of the gate electrode, giving the Schottky barrier height of approximately 0.8V.

After the initial solution, the gate voltage is set to 0, and the Id-Vds characteristic is calculated. First the drain voltage is ramped up to 0.1V. The solutions are obtained using the combined Gummel-Newton block algorithm specified in the following statement:

```
method gummel block newton
```

The algorithm implies that if the solution does not converge in the course of Gummel iterations the program will automatically switch over to the block algorithm where the Poisson and electron continuity equations are solved by the Newton algorithm with the frozen electron temperature; then the electron continuity equation and energy balance equation are solved by the Newton algorithm. If the solution still does not converge the program will switch over to the fully coupled Newton method.

Next the drain voltage is ramped up to 3V. This part of simulation is performed using the Newton method:

```
method newton
```

The results of simulation are saved in a log file to be displayed later with the results of the DD simulation.

The second ATLAS run repeats the first but uses the conventional one carrier Drift Diffusion Model: Poisson's equation and the electron continuity equation are solved self-consistently. The same set of physical models is used, except the energy balance equation. The latter is turned off by not indicating the hcte.el parameter in the models statement. The resulting Id-Vds characteristic is saved in a log file. The statement:

```
output con.band val.band e.velocity
```

is used in both runs to save the conduction and valence band potentials and the electron velocity information to the solution files. Generating the solution files in the energy balance simulation and selecting to plot them will show the velocity overshoot effect.

Results from two runs with the EB and DD models are compared by overlaying the two log files in TONYPLOT.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

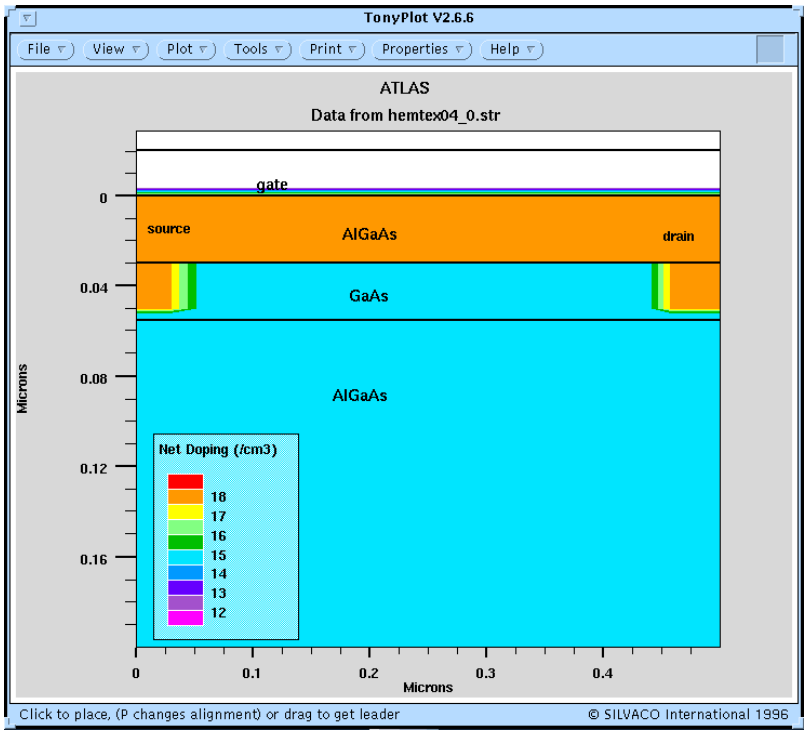


Figure 13.9: HEMT doping profile and material layers

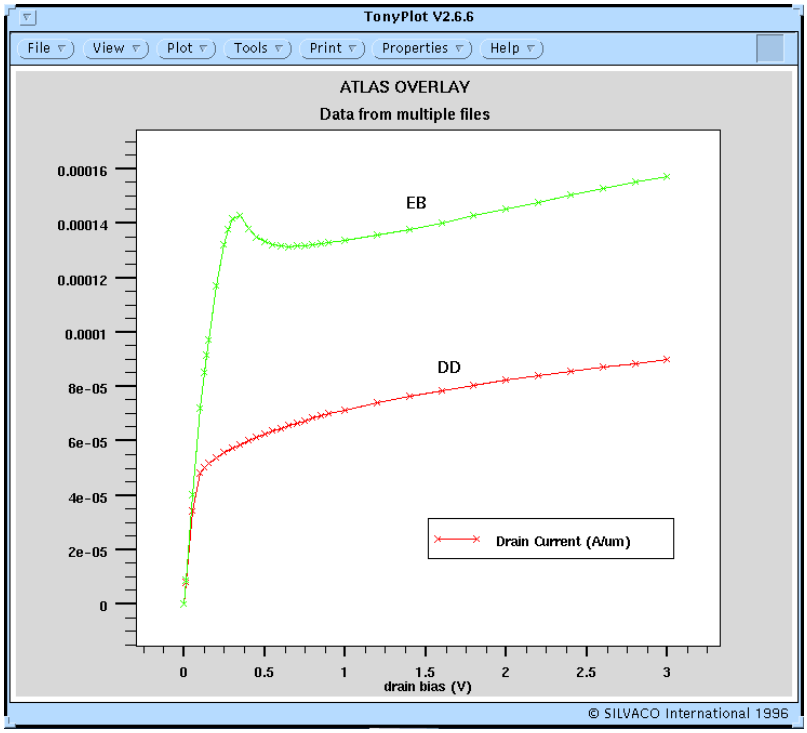


Figure 13.10: Id/Vds are compared using Energy Balance and Drift diffusion Models

Input File hemt/hemtex04.in :

```
1  go atlas
2  # SILVACO International 1996
3  # EB simulation
4  # SECTION 1: Mesh input
5  #
6  mesh        nx=47 ny=47
7  #
8  x.m        n=1    l=0.0    r=1.0
9  x.m        n=4    l=0.08   r=1.0
10 x.m        n=11    l=0.15  r=1.0
11 x.m        n=30   l=0.28   r=1.0
12 x.m        n=42   l=0.4    r=1.0
13 x.m        n=47   l=0.5    r=1.0
14 #
15 y.m        n=1     l=-0.02       r=1.0
16 y.m        n=3     l=0.0         r=1.0
17 y.m        n=10    l=0.025       r=1.0
18 y.m        n=20    l=0.03        r=1.0
19 y.m        n=31    l=0.035       r=1.0
20 y.m        n=37    l=0.05        r=1.0
21 y.m        n=39    l=0.055       r=1.0
22 y.m        n=42    l=0.06        r=1.0
23 y.m        n=47    l=0.200       r=1.05
24 #
25 # SECTION 2: Structure Specification
26 #
27 region     num=1   material=GaAs y.min=0.03 y.max=0.055
28 region     num=2  material=AlGaAs y.max=0.03 x.composition=0.3
29 region     num=3  material=AlGaAs y.min=0.055 x.composition=0.3
30 region     num=4  oxide y.min=-0.02 y.max=0
31
32 elec       num=1  name=source x.min=0.0 x.max=0.0 y.min=0.0 y.max=0.05
33 elec       num=2  name=gate   x.min=0.1 x.max=0.35 y.min=0.0 y.max=0.0
34 elec       num=3  name=drain  x.min=0.5 x.max=0.5 y.min=0.0 y.max=0.05
35
36 #
37 doping     uniform y.min=0 y.max=0.03 n.type conc=1.e18
38 doping     uniform y.min=0.03 n.type conc=1.e15
39 doping     uniform x.min=0.0 x.max=0.05 y.min=0.03 y.max=0.05 n.type
     conc=1.e18
40 doping     uniform x.min=0.45 x.max=0.5 y.min=0.03 y.max=0.05 n.type
     conc=1.e18
41
```

```
42 interface x.min=0 x.max=0.5 y.min=-0.01 y.max=0.005 qf=-1.e12
43
44 # SECTION 3: Material Models
45 #
46
47 material material=GaAs mun=6500 taurel.el=1.e-12 taumob.el=1.e-12
     vsat=1.e7
48 material material=AlGaAs mun=2000 taurel.el=1.e-12 taumob.el=1.e-12
     align=0.6
49
50 model     fldmob    print hcte.el
51 model    material=GaAs  print  evsatmod=1
52
53 contact  number=2 workfun=4.64
54
55 #
56 # SECTION 4: Initial solution
57
58
59 solve init
60 save outf=hemtex04_0.str
61 tonyplot  hemtex04_0.str -set hemtex04_0.set
62
63 method gummel block newton  maxtrap=6 \
64   ir.tol=1.e-20 ix.tol=1.e-20 vsatmod.inc=0.05 carriers=1 electron
65
66 # SECTION 5: Bias gate
67 output con.band val.band e.velocity
68
69 solve vgate=0
70
71 # SECTION 6: Drain ramp
72
73 log outf=hemtex04_eb.log master
74
75 solve
76 solve vdrain=0.01
77
78 solve vdrain=0.05
79 solve vdrain=0.10
80
81 method newton  maxtrap=6  \
82   ir.tol=1.e-20 ix.tol=1.e-20 vsatmod.inc=0.05 carriers=1 electron
83
```

```
84  solve vdrain=0.125
85  solve vdrain=0.15
86  solve vdrain=0.20
87  solve vdrain=0.25 vstep=0.05 vfinal=0.9 name=drain
88
89  solve vdrain=1  vstep=0.2  electr=3 vfinal=1.6
90  save outf=hemtex04_eb.str
91  solve vdrain=1.8  vstep=0.2  electr=3 vfinal=3.0
92
93
94
95
96  go atlas
97  # DD Simulation
98  # SECTION 1: Mesh input
99  #
100 mesh        nx=47 ny=47
101
102 x.m        n=1    l=0.0    r=1.0
103 x.m        n=4    l=0.08   r=1.0
104 x.m        n=11    l=0.15  r=1.0
105 x.m        n=30   l=0.28   r=1.0
106 x.m        n=42   l=0.4    r=1.0
107 x.m        n=47   l=0.5    r=1.0
108
109 #
110 y.m        n=1     l=-0.02       r=1.0
111 y.m        n=3     l=0.0         r=1.0
112 y.m        n=10    l=0.025       r=1.0
113 y.m        n=20    l=0.03        r=1.0
114 y.m        n=31    l=0.035       r=1.0
115 y.m        n=37    l=0.05        r=1.0
116 y.m        n=39    l=0.055       r=1.0
117 y.m        n=42    l=0.06        r=1.0
118 y.m        n=47    l=0.2000      r=1.05
119
120 #
121 # SECTION 2: Structure Specification
122 #
123 region     num=1   material=GaAs y.min=0.03 y.max=0.055
124 region     num=2  material=AlGaAs y.max=0.03 x.composition=0.3
125 region     num=3  material=AlGaAs y.min=0.055 x.composition=0.3
126 region     num=4  oxide y.min=-0.02 y.max=0
```

```
127
128 elec       num=1  name=source x.min=0.0 x.max=0.0 y.min=0.0 y.max=0.05
129 elec       num=2  name=gate   x.min=0.1 x.max=0.35 y.min=0.0 y.max=0.0
130 elec       num=3  name=drain  x.min=0.5 x.max=0.5 y.min=0.0 y.max=0.05
131
132 #
133 doping     uniform y.min=0 y.max=0.03 n.type conc=1.e18
134 doping     uniform y.min=0.03 n.type conc=1.e15
135 doping     uniform x.min=0.0 x.max=0.05 y.min=0.03 y.max=0.05 n.type
      conc=1.e18
136 doping     uniform x.min=0.45 x.max=0.5 y.min=0.03 y.max=0.05 n.type
      conc=1.e18
137
138 interface x.min=0 x.max=0.5 y.min=-0.01 y.max=0.005 qf=-1.e12
139
140 # SECTION 3: Material Models
141 #
142
143 material material=GaAs mun=6500 taurel.el=1.e-12 taumob.el=1.e-12
      vsat=1.e7
144 material material=AlGaAs mun=2000 taurel.el=1.e-12 taumob.el=1.e-12
      align=0.6
145
146 model     fldmob    print
147 model    material=GaAs  print  evsatmod=1
148
149 contact  number=2 workfun=4.64
150
151 #
152 # SECTION 4: Initial solution
153
154 solve init
155
156
157 method gummel  newton  maxtrap=6 \
158   ir.tol=1.e-20 ix.tol=1.e-20 vsatmod.inc=0.05 electron carr=1
159
160 # SECTION 5: Bias gate
161 output con.band val.band e.velocity
162
163 solve vgate=0
164
165 # SECTION 6: Drain ramp
166
```

```
log outf=hemtex04_dd.log master

solve
solve vdrain=0.01

solve vdrain=0.05
solve vdrain=0.10

method newton  maxtrap=6  \
  ir.tol=1.e-20 ix.tol=1.e-20 vsatmod.inc=0.05 electron carr=1

solve vdrain=0.125
solve vdrain=0.15
solve vdrain=0.20
solve vdrain=0.25 vstep=0.05 vfinal=0.9 name=drain

solve vdrain=1  vstep=0.2  electr=3 vfinal=1.6
save outf=hemtex04_dd.str
solve vdrain=1.8  vstep=0.2  electr=3 vfinal=3.0

tonyplot -overlay hemtex04_dd.log hemtex04_eb.log -set hemtex04_log.set

quit

```

13.1.5. hemtex05.in: Recessed Gate Pseudomorphic HEMT DC Characterization

Requires: DEVEDIT/BLAZE

In this example, a pseudomorphic HEMT structure based on the GaAs-AlGaAs-InGaAs-InP material system is constructed using DevEdit. The structure is then passed to ATLAS for electrical testing. The input file consists of the following main portions:

- construction of the device in DEVEDIT
- electrical simulation of a family of Id/Vds curves
- simulation of Id/Vgs characteristics
- basic parameter extraction

The first part of the input file constructs the HEMT geometry, material regions, doping profiles, and electrodes in DEVEDIT. The structure considered is non- planar with a non-rectangular recessed gate. This demonstrates the important capability of DEVEDIT to generate arbitrarily shaped geometries. The device, based on GaAs substrate, is created in DEVEDIT by drawing the device regions in interactive mode and specifying 2D doping distributions. It employs a double channel HEMT concept where the InGaAs channel is sandwiched between two AlGaAs regions. The structure also employs 2 delta- (or pulse-) dopings above and below the channel in both AlGaAs regions. These are modeled by narrow, 10 angstrom thick layers. The delta-doping, usually expressed in terms of planar concentration (cm-2) must be recalculated into bulk doping (cm-3) given the thickness of the layers.

The delta-doped regions play an important role and are typical for modern HEMT technologies. They are used as additional carrier suppliers to the channel and for better control of the threshold voltage and other device parameters. The source and drain cap regions are made of GaAs. In practical applications the source and drain contact alloys often penetrate deep into the structure well below the channel. This is modeled by heavily doped areas under the source and drain in which the vertical doping distribution is assumed to be Gaussian, and the horizontal one is approximated by the complimentary error function. The means and the set of functions for specifying arbitrary 2D doping distributions are provided by DEVEDIT. The composition fractions of AlGaAs (0.22 for Al) and InGaAs (0.78 for Ga) are specified in DEVEDIT as respective region attributes. The mesh was generated automatically by specifying basic mesh constraints and refining it along the x- and/or y-directions in the important areas of the device.

DEVEDIT then generates two types of files: a DEVEDIT input file and the structure file. The input file can be run in DECKBUILD to produce the corresponding structure file and is included here as the first part of the input file. The structure file can be read in directly by ATLAS in the `mesh` statement. Note that the DEVEDIT input file can be edited as any other input file. It is straightforward to change the type and value of the doping associated with each region or resize regions. More importantly, DEVEDIT input files can also be read directly into the graphical user interface of DEVEDIT to provide all the menu options used to construct the structure.

The ATLAS simulation begins from reading in the structure from DEVEDIT. DECKBUILD provides autointerface between DEVEDIT and ATLAS so that the structure produced by DevEdit is transferred to ATLAS without having to indicate the MESH statement (commented out in this example). Without the automatic DEVEDIT/ATLAS interface under DECKBUILD, the MESH statement is needed to load the structure and the mesh.

The first active statements in the ATLAS portion of the input file are the contact, material and model definitions. The gate workfunction is set up in the `contact` statement. The material parameters and physical models are specified in the `material` and `models` statments respectively on a material-by-material basis. Shockley-Read-Hall recombination and electric field dependent mobility models are applied to all the material/regions. For GaAs and AlGaAs regions doping dependent mobilities and recombination parameters (lifetimes) are also applied. Conversely for InGaAs region low field mobilities and carrier lifetimes are explicitly specified in the material statement. The band alignment is defined here using the `align` parameter, which defines the portion of the energy band gap difference applied to the conduction band. This parameter is given in the `material` statements.

The simulation is first performed to obtain the condition of the structure for three different gate biases, Vgs=0, -0.2, and -0.4 V, with the source and drain grounded. The respective states of the structure are saved in three separate solution files. Then the family of three Id-Vds characteristics is calculated in three separate drain voltage sweeps from 0 up to 2 V. Each series of the drain biasing is done after loading the solutions with the respective gate bias. The Id-Vds characteristics are saved in three separate log files. The final conditions of the structure with 2 V on the drain are also saved in separate solution files. The zero bias condition is loaded again and the Id-Vgs characteristic is calculated.

At the end of simulation, parameter extraction statements are used to extract the following parameters using the `extract` feature of DECKBUILD:

* Threshold voltage
* Maximum saturation current (Idss)
* Gate voltage at Id=0.3*Idss

The results are also displayed using TONYPLOT:

* Id/Vds characteristics
* Id/Vgs characteristic

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

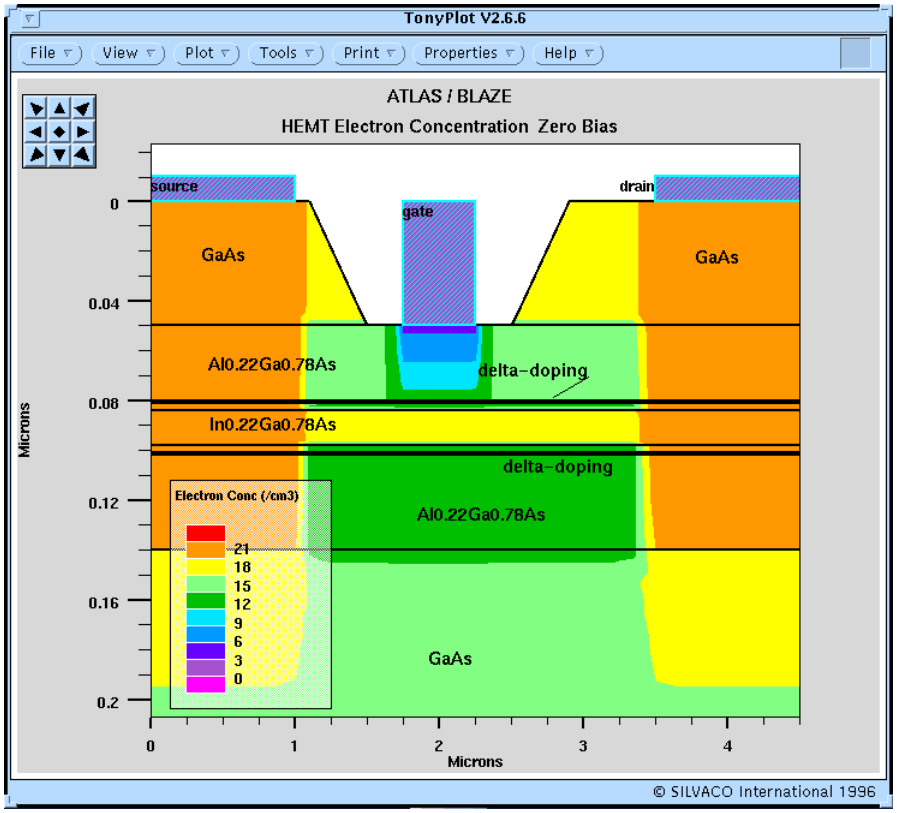


Figure 13.11: Complex pseudomorphic InGaAs HEMT structure and doping defined using DevEdit

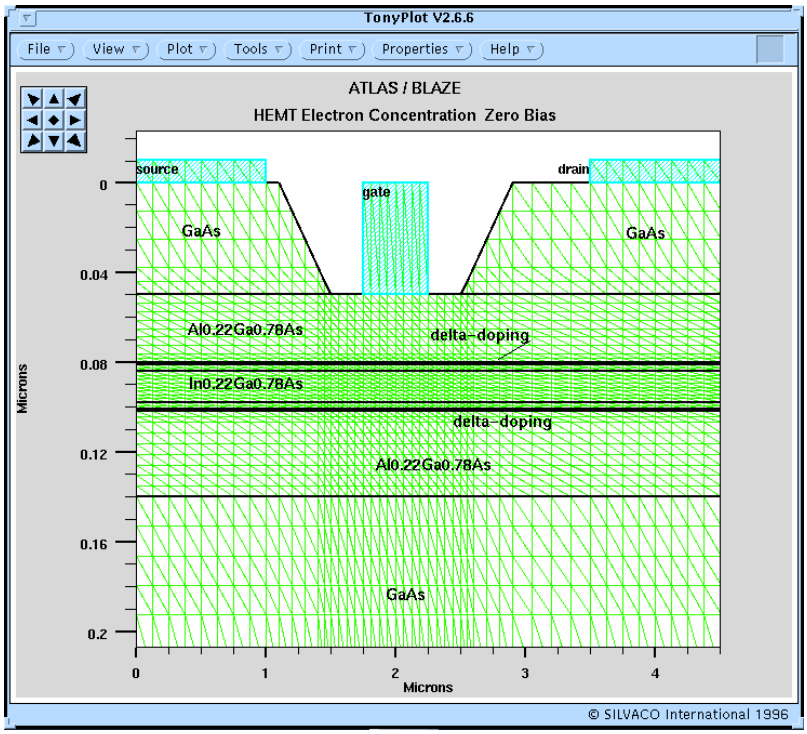


Figure 13.12: Mesh for the recessed HEMT. Note the concentration of mesh in the InGaAs channel

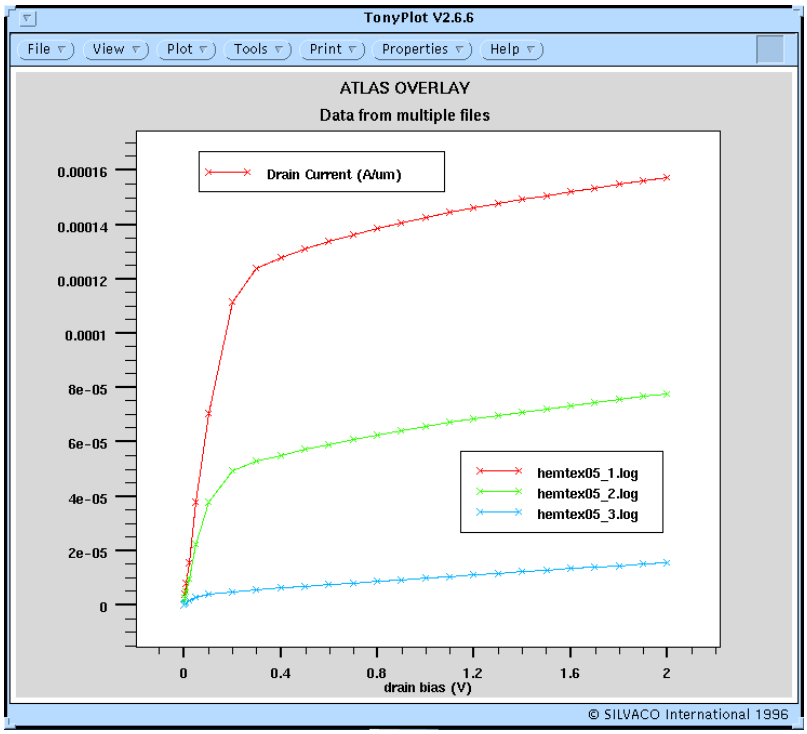


Figure 13.13: Family of Id/Vds curves from the InGaAs HEMT

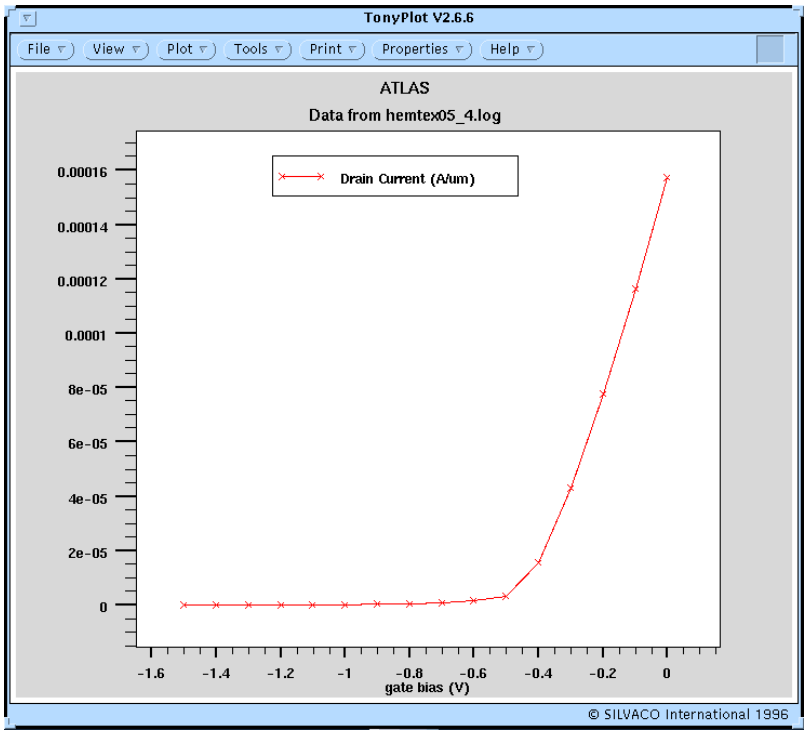


Figure 13.14: Threshold voltage extraction from a PHEMT Id/Vgs curve

Input File hemt/hemtex05.in:

```
go DevEdit
DevEdit version="2.1" library="1.15"

work.area left=0 top=-0.01 right=4.5 bottom=0.5

# SILVACO Library V1.15

region reg=1 mat=GaAs color=0xffb2 pattern=0x9 \
points="0,0 1,0 1.1,0 1.5,0.05 0,0.05 0,0"
#
impurity id=1 region.id=1 imp=Donors color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=2e+18 ref.value=1000000000000 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
constr.mesh region=1 default

region reg=2 mat=GaAs color=0xffb2 pattern=0x9 \
points="2.9,0 3.5,0 4.5,0 4.5,0.05 2.5,0.05 2.9,0"
#
```

```
22 impurity id=1 region.id=2 imp=Donors color=0x906000 \
23 x1=0 x2=0 y1=0 y2=0 \
24 peak.value=2e+18 ref.value=1000000000000 comb.func=Multiply \
25 rolloff.y=both conc.func.y=Constant \
26 rolloff.x=both conc.func.x=Constant
27 #
28 constr.mesh region=2 default
29
30 region reg=3 mat=AlGaAs color=0xffff96 pattern=0x9 \
31 points="4.5,0.05 4.5,0.08 0,0.08 0,0.05 1.5,0.05 1.75,0.05 2.25,0.05
     2.5,0.05 4.5,0.05"
32 #
33 impurity id=1 region.id=3 imp="Composition Fraction X" color=0x906000 \
34 x1=0 x2=0 y1=0 y2=0 \
35 peak.value=0.22 ref.value=0 comb.func=Multiply \
36 rolloff.y=both conc.func.y=Constant \
37 rolloff.x=both conc.func.x=Constant
38 #
39 impurity id=2 region.id=3 imp=Donors color=0x906000 \
40 x1=0 x2=0 y1=0 y2=0 \
41 peak.value=5e+15 ref.value=1000000000000 comb.func=Multiply \
42 rolloff.y=both conc.func.y=Constant \
43 rolloff.x=both conc.func.x=Constant
44 #
45 constr.mesh region=3 default
46
47 region reg=4 name=delta mat=AlGaAs color=0xffff96 pattern=0x9 \
48 points="4.5,0.08 4.5,0.081 0,0.081 0,0.08 4.5,0.08"
49 #
50 impurity id=1 region.id=4 imp="Composition Fraction X" color=0x906000 \
51 x1=0 x2=0 y1=0 y2=0 \
52 peak.value=0.22 ref.value=0 comb.func=Multiply \
53 rolloff.y=both conc.func.y=Constant \
54 rolloff.x=both conc.func.x=Constant
55 #
56 impurity id=2 region.id=4 imp=Donors color=0x906000 \
57 x1=0 x2=0 y1=0 y2=0 \
58 peak.value=8e+18 ref.value=1000000000000 comb.func=Multiply \
59 rolloff.y=both conc.func.y=Constant \
60 rolloff.x=both conc.func.x=Constant
61 #
62 constr.mesh region=4 default
63
```

```
region reg=5 name=spacer mat=AlGaAs color=0xffff96 pattern=0x9 \
points="0,0.081 4.5,0.081 4.5,0.084 0,0.084 0,0.081"
#
impurity id=1 region.id=5 imp="Composition Fraction X" color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=0.22 ref.value=0 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
impurity id=2 region.id=5 imp=Donors color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=5e+15 ref.value=1000000000000 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
constr.mesh region=5 default

region reg=6 mat=InGaAs color=0xffc8c8 pattern=0xa \
points="0,0.084 4.5,0.084 4.5,0.098 0,0.098 0,0.084"
#
impurity id=1 region.id=6 imp="Composition Fraction X" color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=0.78 ref.value=0 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
impurity id=2 region.id=6 imp=Donors color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=5e+15 ref.value=1000000000000 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
constr.mesh region=6 default

region reg=7 mat=AlGaAs color=0xffff96 pattern=0x9 \
points="0,0.098 4.5,0.098 4.5,0.101 0,0.101 0,0.098"
#
impurity id=1 region.id=7 imp=Acceptors color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=150000000000000 ref.value=1000000000000 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
```

```
107 impurity id=2 region.id=7 imp="Composition Fraction X" color=0x906000 \
108 x1=0 x2=0 y1=0 y2=0 \
109 peak.value=0.22 ref.value=0 comb.func=Multiply \
110 rolloff.y=both conc.func.y=Constant \
111 rolloff.x=both conc.func.x=Constant
112 #
113 constr.mesh region=7 default
114
115 region reg=8 mat=AlGaAs color=0xffff96 pattern=0x9 \
116 points="0,0.101 4.5,0.101 4.5,0.102 0,0.102 0,0.101"
117 #
118 impurity id=1 region.id=8 imp="Composition Fraction X" color=0x906000 \
119 x1=0 x2=0 y1=0 y2=0 \
120 peak.value=0.22 ref.value=0 comb.func=Multiply \
121 rolloff.y=both conc.func.y=Constant \
122 rolloff.x=both conc.func.x=Constant
123 #
124 impurity id=2 region.id=8 imp=Donors color=0x906000 \
125 x1=0 x2=0 y1=0 y2=0 \
126 peak.value=2e+18 ref.value=1000000000000 comb.func=Multiply \
127 rolloff.y=both conc.func.y=Constant \
128 rolloff.x=both conc.func.x=Constant
129 #
130 constr.mesh region=8 default
131
132 region reg=9 mat=AlGaAs color=0xffff96 pattern=0x9 \
133 points="0,0.102 4.5,0.102 4.5,0.14 0,0.14 0,0.102"
134 #
135 impurity id=1 region.id=9 imp=Acceptors color=0x906000 \
136 x1=0 x2=0 y1=0 y2=0 \
137 peak.value=100000000000000 ref.value=1000000000000 comb.func=Multiply \
138 rolloff.y=both conc.func.y=Constant \
139 rolloff.x=both conc.func.x=Constant
140 #
141 impurity id=2 region.id=9 imp="Composition Fraction X" color=0x906000 \
142 x1=0 x2=0 y1=0 y2=0 \
143 peak.value=0.22 ref.value=0 comb.func=Multiply \
144 rolloff.y=both conc.func.y=Constant \
145 rolloff.x=both conc.func.x=Constant
146 #
147 constr.mesh region=9 default
148
149 region reg=10 name=GaAs mat=AlGaAs color=0xffb2 pattern=0x9 \
```

```
150 points="0,0.14 4.5,0.14 4.5,0.5 0,0.5 0,0.14"
151 #
152 impurity id=1 region.id=10 imp=Acceptors color=0x906000 \
153 x1=0 x2=0 y1=0 y2=0 \
154 peak.value=100000000000000 ref.value=1000000000000 comb.func=Multiply \
155 rolloff.y=both conc.func.y=Constant \
156 rolloff.x=both conc.func.x=Constant
157 #
158 constr.mesh region=10 default
159
160 region reg=11 name=source mat=Gold elec.id=1 work.func=0 color=0xe5ff
      pattern=0xb \
161 points="0,0 0,-0.01 1,-0.01 1,0 0,0"
162 #
163 constr.mesh region=11 default
164
165 region reg=12 name=drain mat=Gold elec.id=2 work.func=0 color=0xe5ff pat-
      tern=0xb \
166 points="3.5,0 3.5,-0.01 4.5,-0.01 4.5,0 3.5,0"
167 #
168 constr.mesh region=12 default
169
170 region reg=13 name=gate mat=Gold elec.id=3 work.func=0 color=0xe5ff pat-
      tern=0xb \
171 points="1.75,0.05 1.75,0 2.25,0 2.25,0.05 1.75,0.05"
172 #
173 constr.mesh region=13 default
174
175
176 impurity id=1 imp=Donors color=0x906000 \
177 x1=0 x2=1 y1=0 y2=0.08 \
178 peak.value=5e+21 ref.value=1e+20 comb.func=Multiply \
179 rolloff.y=both conc.func.y="Gaussian (Dist)" conc.param.y=0.048 \
180 rolloff.x=both conc.func.x="Error Function" conc.param.x=0.02
181 impurity id=2 imp=Donors color=0x906000 \
182 x1=3.5 x2=4.5 y1=0 y2=0.08 \
183 peak.value=5e+21 ref.value=1e+20 comb.func=Multiply \
184 rolloff.y=both conc.func.y="Gaussian (Dist)" conc.param.y=0.048 \
185 rolloff.x=both conc.func.x="Error Function" conc.param.x=0.02
186
187 # Set Meshing Parameters
188 #
189 base.mesh height=0.1 width=0.25
190 #
```

```
191 bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=5e-05
      line.straightening=1 align.points when=automatic
192 #
193 imp.refine min.spacing=0.02
194 #
195 constr.mesh max.angle=90 max.ratio=3000 max.height=1 \
196 max.width=1 min.height=0.0001 min.width=0.0001
197 #
198 constr.mesh type=Semiconductor default
199 #
200 constr.mesh type=Insulator default
201 #
202 constr.mesh type=Metal default
203 #
204 constr.mesh type=Other default
205 #
206 constr.mesh region=1 default
207 #
208 constr.mesh region=2 default
209 #
210 constr.mesh region=3 default
211 #
212 constr.mesh region=4 default
213 #
214 constr.mesh region=5 default
215 #
216 constr.mesh region=6 default
217 #
218 constr.mesh region=7 default
219 #
220 constr.mesh region=8 default
221 #
222 constr.mesh region=9 default
223 #
224 constr.mesh region=10 default
225 #
226 constr.mesh region=11 default
227 #
228 constr.mesh region=12 default
229 #
230 constr.mesh region=13 default
231 #
232 # Perform mesh operations
```

```
#
Mesh Mode=MeshBuild
refine mode=y x1=0.068 y1=0.0123 x2=1.48 y2=0.038
refine mode=y x1=2.492 y1=0.0106 x2=4.402 y2=0.044
refine mode=y x1=0.106 y1=0.0089 x2=1.511 y2=0.0414
refine mode=y x1=2.515 y1=0.0046 x2=4.357 y2=0.0405
refine mode=x x1=1.571 y1=0.0568 x2=2.417 y2=0.334
refine mode=x x1=0.084 y1=0.0063 x2=0.891 y2=0.3477
refine mode=x x1=3.602 y1=0.0055 x2=4.41 y2=0.3383
refine mode=y x1=0.038 y1=0.0414 x2=1.563 y2=0.0457
refine mode=y x1=2.454 y1=0.0397 x2=4.448 y2=0.044
refine mode=y x1=0.076 y1=0.0551 x2=4.455 y2=0.2887
refine mode=y x1=0.031 y1=0.0524 x2=4.425 y2=0.074
refine mode=y x1=0.068 y1=0.0516 x2=4.44 y2=0.0761
refine mode=y x1=0.023 y1=0.0768 x2=4.463 y2=0.0787
refine mode=y x1=0.084 y1=0.106 x2=4.47 y2=0.1829
refine mode=y x1=0.061 y1=0.1044 x2=4.47 y2=0.1344
refine mode=y x1=0.046 y1=0.1036 x2=4.433 y2=0.1053
refine mode=y x1=0.061 y1=0.0862 x2=4.433 y2=0.0966
refine mode=y x1=0.038 y1=0.0856 x2=4.433 y2=0.0961
refine mode=x x1=1.556 y1=0.0499 x2=2.462 y2=0.2322

imp.refine min.spacing=0.02

constr.mesh max.angle=90 max.ratio=3000 max.height=1 \
max.width=1 min.height=0.0001 min.width=0.0001
#
constr.mesh type=Semiconductor default
#
constr.mesh type=Insulator default
#
constr.mesh type=Metal default
#
constr.mesh type=Other default


base.mesh height=0.1 width=0.25

bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=5e-05
      line.straightening=1 align.Points when=automatic


structure outf=hemtex04_A.str
```

```
275
276
277 go atlas
278
279 title  GaAs-AlGaAs-InGaAs Pseudomorphic HEMT simulation
280
281 # Load the structure generated in DevEdit...
282 # While running in the deckbuild after DevEdit Input File the structure is
283 # transfered automatically: no mesh statement is needed, comment it out
284
285 #mesh infile=hemtex05_A.str master.in
286
287
288
289 # Define the workfunction for the gate contact
290
291 contact     name=gate   workfunction=3.0
292
293 # Define material parameters on material-by-material basis
294
295 material material=GaAs   align=0.6
296 material material=AlGaAs align=0.6
297 material material=InGaAs align=0.6 mun0=12000 mup0=2000 vsat=2.e7 \
298                          taun0=1.e-8 taup0=1.e-8
299
300 # Define physical models on material-by material basis
301
302 models material=GaAs   consrh conmob fldmob evsatmod=0 print
303 models material=AlGaAs consrh conmob fldmob evsatmod=0 print
304 models material=InGaAs srh           fldmob evsatmod=0 print
305
306 # Initial solution
307
308 solve    init
309 save outf=hemtex05_1.str
310 tonyplot  hemtex05_1.str -set hemtex05_1.set
311
312
313 # Include bands potential, and current flowlines into output
314
315 output con.band val.band flowlines
316
317
```

```
# Apply a set of biases at the gate and save solutions

solve vgate= 0    outf=hemtex05_A_0.out
solve vgate=-0.2  outf=hemtex05_Ag-02.out
solve vgate=-0.4  outf=hemtex05_Ag-04.out

# Calculate ID-VD characteristic at zero gate bias

load inf=hemtex05_A_0.out

log outf=hemtex05_1.log


solve outf=hemtex05_A_0.out master
solve vdrain=0.02
solve vdrain=0.05
solve vdrain=0.1 vstep=0.1 vfinal=2.0 name=drain
save outf=hemtex05_Ad2.out master

# Calculate ID-VD characteristic at VG=-0.2

load inf=hemtex05_Ag-02.out
log outf=hemtex05_2.log

solve prev
solve vdrain=0.02
solve vdrain=0.05
solve vdrain=0.1 vstep=0.1 vfinal=2.0 name=drain
save outf=hemtex05_Ad2g-02.out master

# Calculate ID-VD characteristic at VG=-0.4

load inf=hemtex05_Ag-04.out
log outf=hemtex05_3.log

solve prev
solve vdrain=0.02
solve vdrain=0.05
method   itlimit=10 trap atrap=0.5 maxtrap=8
solve vdrain=0.1 vstep=0.1 vfinal=2.0 name=drain
save outf=hemtex05_Ad2g-04.out master


```

```
# Calculate ID-VG characteristics at VD=2.0 V
# Simultaneously apply small signal perturbation to get AC parameters

load inf=hemtex05_Ad2.out master

log outf=hemtex05_4.log master

solve vgate=0 vstep=-0.1 vfinal=-1.5 name=gate ac freq=1e6
save outf=hemtex05_Ad2g_15.out master


extract init inf="hemtex05_4.log"

extract name="Vt" (xintercept(maxslope(curve(v."gate",i."drain"))))

# Maximum saturation current
extract name="Idss" max(i."drain")

# Gate voltage Vg@Idss*0.3 at Id=0.3Idss
extract name="Vgs@0.3Idss" x.val from curve (v."gate", i."drain") where
  y.val=$"Idss"*0.3

# Maximum gate-source capacitance
extract name="Cgs_max" max(abs(c."gate""source"))

# Gate-source capacitance at Vg=0
extract name="Cgs_Vgs0" y.val from curve (v."gate",
  abs(c."gate""source")) where x.val=0.0

# Gate-source capacitance at Id=0.3Idss
extract name="Cgs_Vgs@0.3Idss" y.val from curve (v."gate",
  abs(c."gate""source")) where x.val=$"Vgs@0.3Idss"

# Maximum transconductance
extract name="Gm_max" max(abs(g."drain""gate"))

# Transconductance at Vg=0
extract name="Gm_Vgs0" y.val from curve (v."gate", abs(g."drain""gate"))
  where x.val=0.0

# Transconductance at Id=0.3Idss
extract name="Gm_Vgs@0.3Idss" y.val from curve (v."gate",
  abs(g."drain""gate")) where x.val=$"Vgs@0.3Idss"
# Displaying the results
```

```
400
401 # Plot ID-VD characteristics
402
403 tonyplot -overlay hemtex05_1.log hemtex05_2.log hemtex05_3.log -set
      hemtex05_log.set
404
405
406 # Plot ID-VG characteristic
407 tonyplot hemtex05_4.log -set hemtex05_log1.set
408
409
410
411
412
413
```

13.1.6. hemtex06.in: PHEMT High Frequency Analysis

Requires: DEVEDIT/BLAZE

- AC parameters as a function of the gate bias
- frequency domain AC calculation
- calculation of s-parameters

This example is based on the previous example (see that text for details on the structure).

Once the structure is loaded into BLAZE, a small signal AC perturbation is applied at a frequency of 1 MHz to calculate the AC parameters(conductances and capacitances) as functions of Vgs. This also allows the user to obtain the cutoff frequency using the low frequency approximation through TONYPLOT or by using the extraction feature of DECKBUILD. Currents, voltages, and AC parameters are saved in the log file, and internal structure information is saved in a solution file for the final bias condition of Vgs=-1.5V, Vds=2 V.

The next step is a full frequency domain AC simulation when the **Y** parameter matrix (conductances and capacitances) are calculated as functions of frequency from 10 Hz up to 50 GHz. These results are stored in a log file. Additionally, with the specification of `s.param` on the log statement, the s-parameters are stored to the file.

At the end of simulation, parameter extraction statements are used to extract the following parameters using the `extract` feature of DECKBUILD:

```
* Maximum gate-source capacitance
* Gate-source capacitance at Vgs=0
* Gate-source capacitance at Id=0.3*Idss
* Maximum transconductance
* Transconductance at Vgs=0
* Transconductance at Id=0.3*Idss
* Maximum cutoff frequency
* Cutoff frequency at Vgs=0
* Cutoff frequency at Id=0.3*Idss
* Maximum AC current gain f=1MHz
* AC current gain at Vgs=0 f=1MHz
* AC current gain at Id=0.3*Idss f=1MHz
```

The results are also displayed using TonyPlot:

* Transconductance versus gate bias characteristic
* s-parameters: S12 & S21 in polar coordinates
* s-parameters: S11 & S22 on a Smith Chart

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into the DECKBUILD, select the **run** button to execute the example.

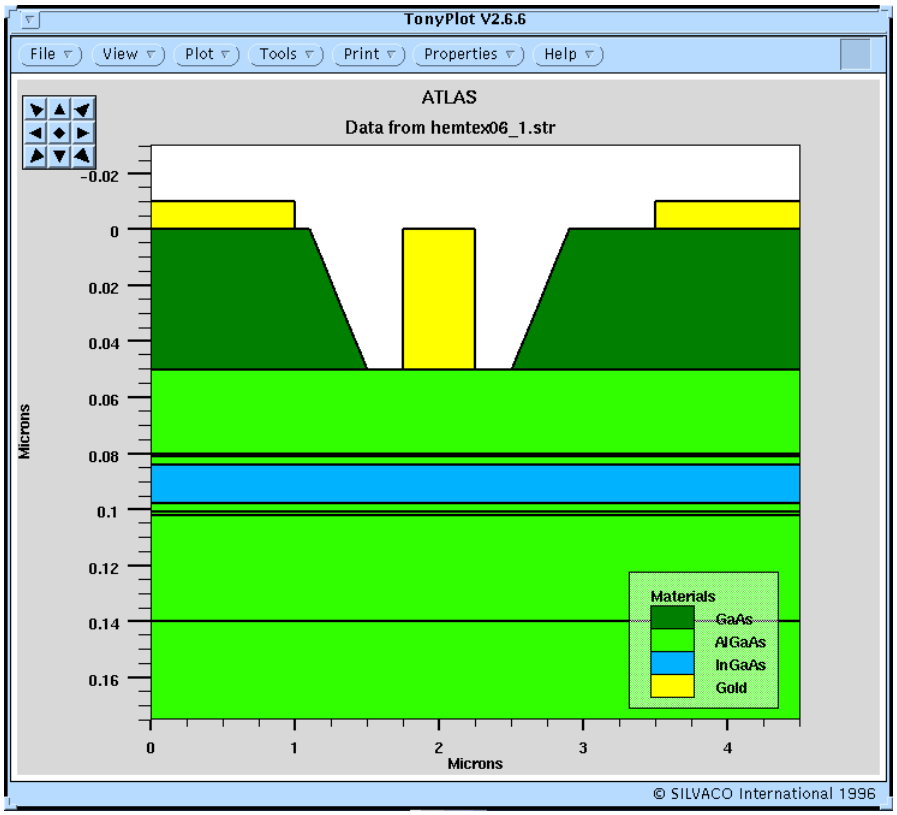


Figure 13.15: PHEMT structure as material layers defined in DevEdit

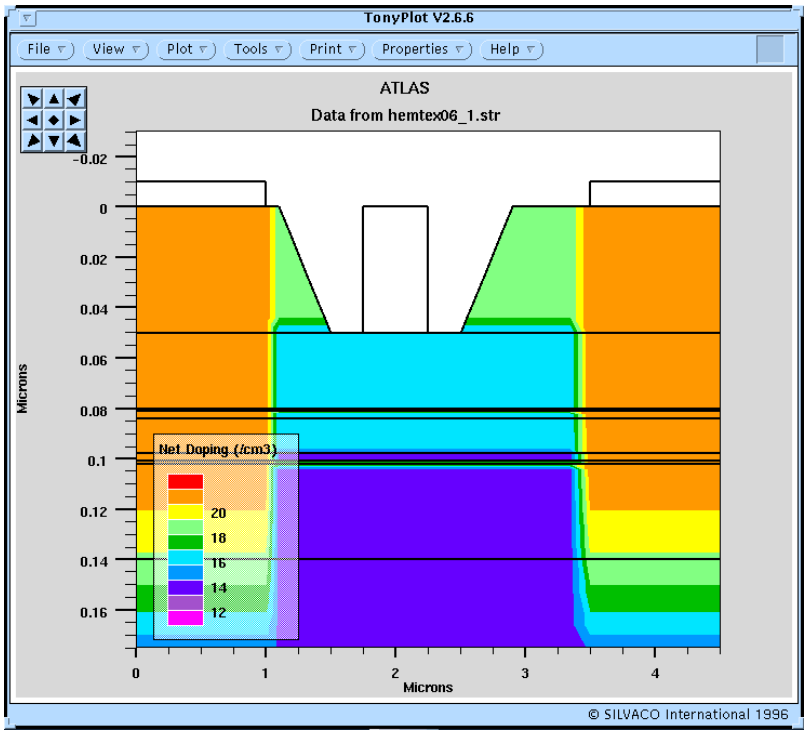


Figure 13.16: PHEMT Doping profile

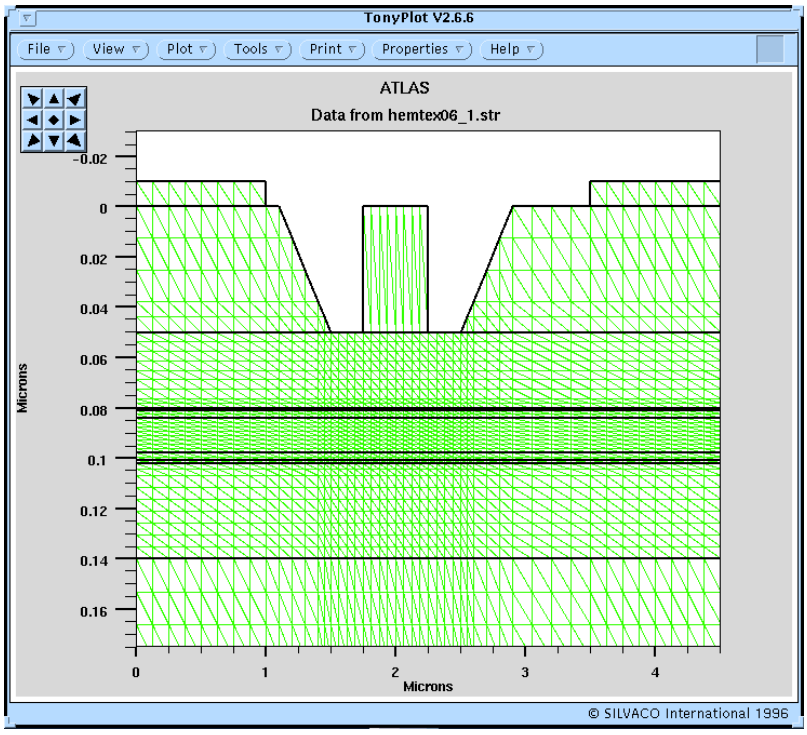


Figure 13.17: PHEMT mesh with mesh concentrated in the InGaAs channel

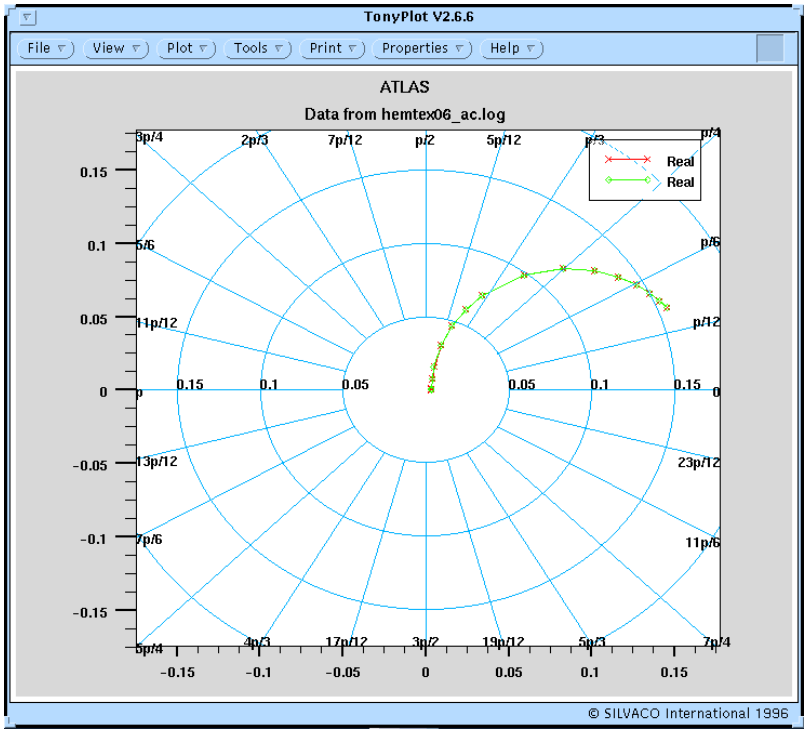


Figure 13.18: Polar plot of PHEMT s-parameters

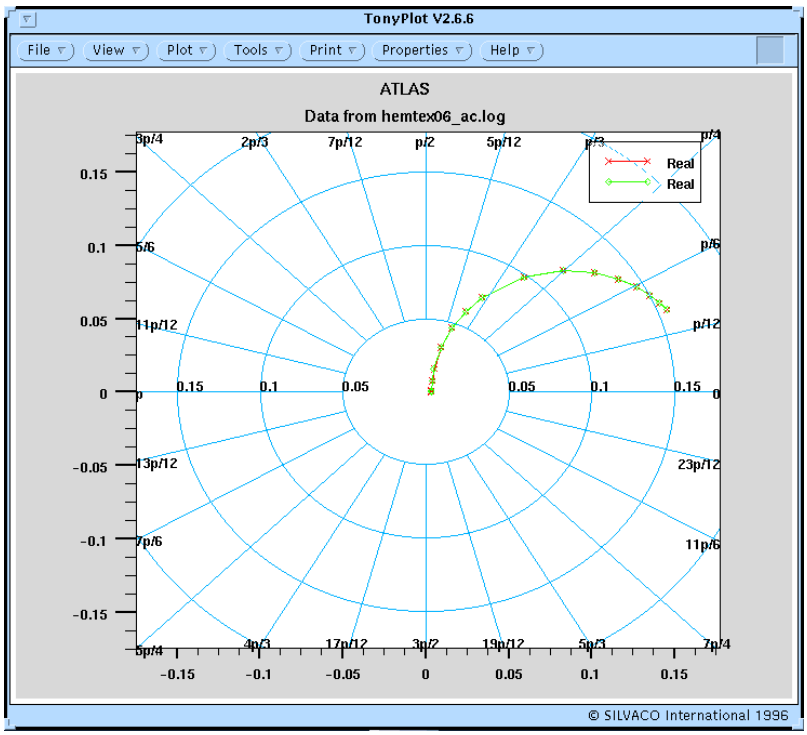


Figure 13.19: Smith Chart of S11 for the PHEMT

Input File hemt/hemtex06.in:

```
1   go DevEdit
2   DevEdit version="2.1" library="1.15"
3
4   work.area left=0 top=-0.01 right=4.5 bottom=0.5
5
6   # SILVACO Library V1.15
7
8   region reg=1 mat=GaAs color=0xffb2 pattern=0x9 \
9   points="0,0 1,0 1.1,0 1.5,0.05 0,0.05 0,0"
10  #
11  impurity id=1 region.id=1 imp=Donors color=0x906000 \
12  x1=0 x2=0 y1=0 y2=0 \
13  peak.value=2e+18 ref.value=1000000000000 comb.func=Multiply \
14  rolloff.y=both conc.func.y=Constant \
15  rolloff.x=both conc.func.x=Constant
16  #
17  constr.mesh region=1 default
18
19  region reg=2 mat=GaAs color=0xffb2 pattern=0x9 \
20  points="2.9,0 3.5,0 4.5,0 4.5,0.05 2.5,0.05 2.9,0"
21  #
22  impurity id=1 region.id=2 imp=Donors color=0x906000 \
23  x1=0 x2=0 y1=0 y2=0 \
24  peak.value=2e+18 ref.value=1000000000000 comb.func=Multiply \
25  rolloff.y=both conc.func.y=Constant \
26  rolloff.x=both conc.func.x=Constant
27  #
28  constr.mesh region=2 default
29
30  region reg=3 mat=AlGaAs color=0xffff96 pattern=0x9 \
31  points="4.5,0.05 4.5,0.08 0,0.08 0,0.05 1.5,0.05 1.75,0.05 2.25,0.05
      2.5,0.05 4.5,0.05"
32  #
33  impurity id=1 region.id=3 imp="Composition Fraction X" color=0x906000 \
34  x1=0 x2=0 y1=0 y2=0 \
35  peak.value=0.22 ref.value=0 comb.func=Multiply \
36  rolloff.y=both conc.func.y=Constant \
37  rolloff.x=both conc.func.x=Constant
38  #
39  impurity id=2 region.id=3 imp=Donors color=0x906000 \
40  x1=0 x2=0 y1=0 y2=0 \
41  peak.value=5e+15 ref.value=1000000000000 comb.func=Multiply \
```

```
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
constr.mesh region=3 default

region reg=4 name=delta mat=AlGaAs color=0xffff96 pattern=0x9 \
points="4.5,0.08 4.5,0.081 0,0.081 0,0.08 4.5,0.08"
#
impurity id=1 region.id=4 imp="Composition Fraction X" color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=0.22 ref.value=0 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
impurity id=2 region.id=4 imp=Donors color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=8e+18 ref.value=1000000000000 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
constr.mesh region=4 default

region reg=5 name=spacer mat=AlGaAs color=0xffff96 pattern=0x9 \
points="0,0.081 4.5,0.081 4.5,0.084 0,0.084 0,0.081"
#
impurity id=1 region.id=5 imp="Composition Fraction X" color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=0.22 ref.value=0 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
impurity id=2 region.id=5 imp=Donors color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=5e+15 ref.value=1000000000000 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
constr.mesh region=5 default

region reg=6 mat=InGaAs color=0xffc8c8 pattern=0xa \
points="0,0.084 4.5,0.084 4.5,0.098 0,0.098 0,0.084"
#
impurity id=1 region.id=6 imp="Composition Fraction X" color=0x906000 \
```

```
x1=0 x2=0 y1=0 y2=0 \
peak.value=0.78 ref.value=0 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
impurity id=2 region.id=6 imp=Donors color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=5e+15 ref.value=1000000000000 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
constr.mesh region=6 default

region reg=7 mat=AlGaAs color=0xffff96 pattern=0x9 \
points="0,0.098 4.5,0.098 4.5,0.101 0,0.101 0,0.098"
#
impurity id=1 region.id=7 imp=Acceptors color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=150000000000000 ref.value=1000000000000 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
impurity id=2 region.id=7 imp="Composition Fraction X" color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=0.22 ref.value=0 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
constr.mesh region=7 default

region reg=8 mat=AlGaAs color=0xffff96 pattern=0x9 \
points="0,0.101 4.5,0.101 4.5,0.102 0,0.102 0,0.101"
#
impurity id=1 region.id=8 imp="Composition Fraction X" color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=0.22 ref.value=0 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
impurity id=2 region.id=8 imp=Donors color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=2e+18 ref.value=1000000000000 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
```

```
rolloff.x=both conc.func.x=Constant
#
constr.mesh region=8 default

region reg=9 mat=AlGaAs color=0xffff96 pattern=0x9 \
points="0,0.102 4.5,0.102 4.5,0.14 0,0.14 0,0.102"
#
impurity id=1 region.id=9 imp=Acceptors color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=100000000000000 ref.value=1000000000000 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
impurity id=2 region.id=9 imp="Composition Fraction X" color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=0.22 ref.value=0 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
constr.mesh region=9 default

region reg=10 name=GaAs mat=AlGaAs color=0xffb2 pattern=0x9 \
points="0,0.14 4.5,0.14 4.5,0.5 0,0.5 0,0.14"
#
impurity id=1 region.id=10 imp=Acceptors color=0x906000 \
x1=0 x2=0 y1=0 y2=0 \
peak.value=100000000000000 ref.value=1000000000000 comb.func=Multiply \
rolloff.y=both conc.func.y=Constant \
rolloff.x=both conc.func.x=Constant
#
constr.mesh region=10 default

region reg=11 name=source mat=Gold elec.id=1 work.func=0 color=0xe5ff
  pattern=0xb \
points="0,0 0,-0.01 1,-0.01 1,0 0,0"
#
constr.mesh region=11 default

region reg=12 name=drain mat=Gold elec.id=2 work.func=0 color=0xe5ff pat-
  tern=0xb \
points="3.5,0 3.5,-0.01 4.5,-0.01 4.5,0 3.5,0"
#
constr.mesh region=12 default

```

```
region reg=13 name=gate mat=Gold elec.id=3 work.func=0 color=0xe5ff pat-
  tern=0xb \
points="1.75,0.05 1.75,0 2.25,0 2.25,0.05 1.75,0.05"
#
constr.mesh region=13 default


impurity id=1 imp=Donors color=0x906000 \
x1=0 x2=1 y1=0 y2=0.08 \
peak.value=5e+21 ref.value=1e+20 comb.func=Multiply \
rolloff.y=both conc.func.y="Gaussian (Dist)" conc.param.y=0.048 \
rolloff.x=both conc.func.x="Error Function" conc.param.x=0.02
impurity id=2 imp=Donors color=0x906000 \
x1=3.5 x2=4.5 y1=0 y2=0.08 \
peak.value=5e+21 ref.value=1e+20 comb.func=Multiply \
rolloff.y=both conc.func.y="Gaussian (Dist)" conc.param.y=0.048 \
rolloff.x=both conc.func.x="Error Function" conc.param.x=0.02

# Set Meshing Parameters
#
base.mesh height=0.1 width=0.25
#
bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=5e-05
  line.straightening=1 align.points when=automatic
#
imp.refine min.spacing=0.02
#
constr.mesh max.angle=90 max.ratio=3000 max.height=1 \
max.width=1 min.height=0.0001 min.width=0.0001
#
constr.mesh type=Semiconductor default
#
constr.mesh type=Insulator default
#
constr.mesh type=Metal default
#
constr.mesh type=Other default
#
constr.mesh region=1 default
#
constr.mesh region=2 default
#
constr.mesh region=3 default
#
```

```
212 constr.mesh region=4 default
213 #
214 constr.mesh region=5 default
215 #
216 constr.mesh region=6 default
217 #
218 constr.mesh region=7 default
219 #
220 constr.mesh region=8 default
221 #
222 constr.mesh region=9 default
223 #
224 constr.mesh region=10 default
225 #
226 constr.mesh region=11 default
227 #
228 constr.mesh region=12 default
229 #
230 constr.mesh region=13 default
231 #
232 # Perform mesh operations
233 #
234 Mesh Mode=MeshBuild
235 refine mode=y x1=0.068 y1=0.0123 x2=1.48 y2=0.038
236 refine mode=y x1=2.492 y1=0.0106 x2=4.402 y2=0.044
237 refine mode=y x1=0.106 y1=0.0089 x2=1.511 y2=0.0414
238 refine mode=y x1=2.515 y1=0.0046 x2=4.357 y2=0.0405
239 refine mode=x x1=1.571 y1=0.0568 x2=2.417 y2=0.334
240 refine mode=x x1=0.084 y1=0.0063 x2=0.891 y2=0.3477
241 refine mode=x x1=3.602 y1=0.0055 x2=4.41 y2=0.3383
242 refine mode=y x1=0.038 y1=0.0414 x2=1.563 y2=0.0457
243 refine mode=y x1=2.454 y1=0.0397 x2=4.448 y2=0.044
244 refine mode=y x1=0.076 y1=0.0551 x2=4.455 y2=0.2887
245 refine mode=y x1=0.031 y1=0.0524 x2=4.425 y2=0.074
246 refine mode=y x1=0.068 y1=0.0516 x2=4.44 y2=0.0761
247 refine mode=y x1=0.023 y1=0.0768 x2=4.463 y2=0.0787
248 refine mode=y x1=0.084 y1=0.106 x2=4.47 y2=0.1829
249 refine mode=y x1=0.061 y1=0.1044 x2=4.47 y2=0.1344
250 refine mode=y x1=0.046 y1=0.1036 x2=4.433 y2=0.1053
251 refine mode=y x1=0.061 y1=0.0862 x2=4.433 y2=0.0966
252 refine mode=y x1=0.038 y1=0.0856 x2=4.433 y2=0.0961
253 refine mode=x x1=1.556 y1=0.0499 x2=2.462 y2=0.2322
254
```

```
imp.refine min.spacing=0.02

constr.mesh max.angle=90 max.ratio=3000 max.height=1 \
max.width=1 min.height=0.0001 min.width=0.0001
#
constr.mesh type=Semiconductor default
#
constr.mesh type=Insulator default
#
constr.mesh type=Metal default
#
constr.mesh type=Other default


base.mesh height=0.1 width=0.25

bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=5e-05
  line.straightening=1 align.Points when=automatic


structure outf=hemtex06_A.str


go atlas

title  GaAs-AlGaAs-InGaAs Pseudomorphic HEMT simulation

# Load the structure generated in DevEdit...
# While running in the deckbuild after DevEdit Input File the structure is
# transfered automatically: no mesh statement is needed, comment it out

#mesh infile=hemtex06_A.str master.in



# Define the workfunction for the gate contact

contact     name=gate   workfunction=3.0

# Define material parameters on material-by-material basis

material material=GaAs   align=0.6
material material=AlGaAs align=0.6
```

```
material material=InGaAs align=0.6 mun0=12000 mup0=2000 vsat=2.e7 \
                         taun0=1.e-8 taup0=1.e-8

# Define physical models on material-by material basis

models material=GaAs   consrh conmob fldmob print
models material=AlGaAs consrh conmob fldmob print
models material=InGaAs srh           fldmob print

# Initial solution

solve    init
save outf=hemtex06_1.str
tonyplot  hemtex06_1.str


# Include bands potential, and current flowlines into output

output con.band val.band flowlines


# Apply a set of biases at the gate and save solutions

solve vgate= 0

# Frequency domain AC simulation.

log outf=hemtex06_ac.log master s.params inport=gate outport=drain
  width=50

solve ac freq=10 fstep=10 mult.f nfstep=7
solve ac freq=1e9
solve ac freq=2e9 fstep=2e9 nfstep=3
solve ac freq=1e10 fstep=5e9 nfstep=8

# Displaying the results

# Plot S-parameters: S12 & S21 in polar coordinates
tonyplot hemtex06_ac.log -set hemtex06_2.set

# Plot S-parameters: S11 & S22 in the form of the Smith Chart

tonyplot  hemtex06_ac.log -set hemtex06_3.set
```

```
339
340 quit
341
342
343
344
```

This page is intentionally left blank

14.1. QUANTUM: Device Simulation with Quantum Mechanics

14.1.1 quantumex01.in: CV Analysis Of Thin Gate Oxide PMOS Capacitor

Requires: PISCES/QUANTUM

This example demonstrates the CV characteristics of a N-type substrate MOS capacitor with a nominal 30A gate oxide. The example shows:

- Calculation of CV curves using ac Analysis
- Use of the Quantum moments quantum model
- Difference between classical and quantum gate thickness
- Classical and Quantized electron density

A simple 1D MOS capacitor is created in this example with a 30A gate oxide thickness. An ac analysis is performed on the device first with the `classical` simulation and then with the `quantum` model swtiched on.

The quantum model is activated by using the `quantum` switch in the models statement for electrons. Note that this activates a quantum moments model which uses a quantum temperature calculation. This quantum temperature can be viewed in the structure files. Also, the cutline created in TONYPLOT can be used to examine the classical and quantum electron concenration differences in the structure.

In accumulation, the channel carrier concentration is changed by the addition of quantum mechanics. The peak is not so high and the electron spreads more deeply into the substrate. The difference in gate capacitance is noticable on the `CV` plot. Since gate oxide thickness is often measured using CV techniques this leads to erroneous results if `QM` effects are not accounted for. `Extract` statements are used to show the calculated Tox from the classical and `QM CV` curves.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

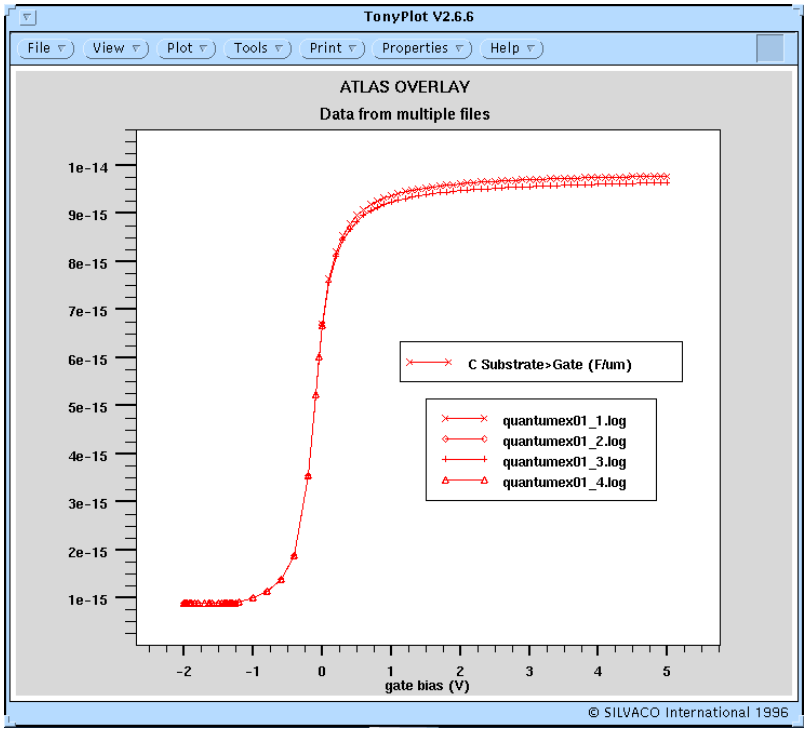


Figure 14.1: Comparison of NMOS CV curve using Quantum (plus) and Classical (x) models. An offset is seen in accumulation

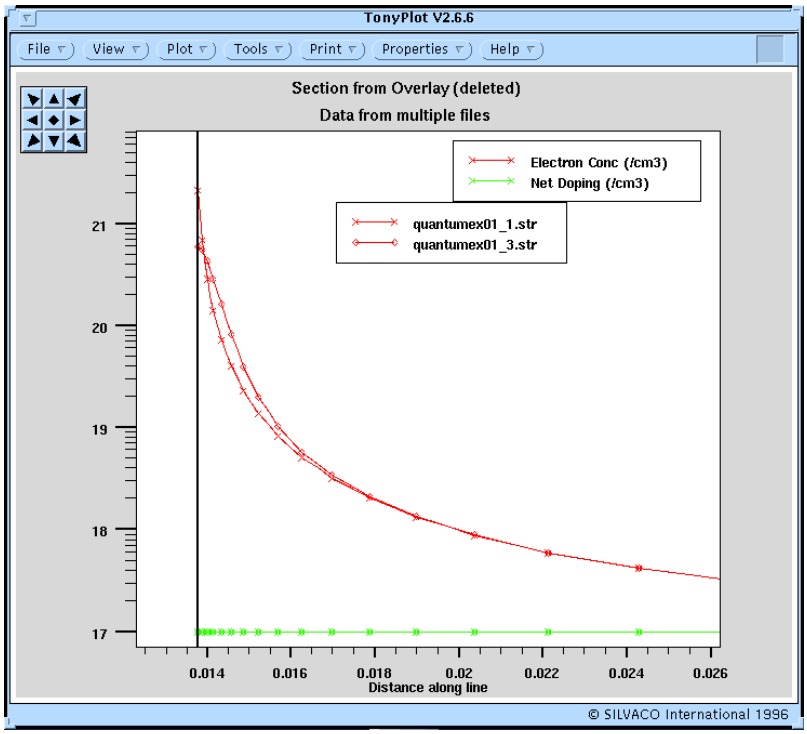


Figure 14.2: Electron concentration in the inversion layer using classical (x) and quantum (o) statistics

Input File quantum/quantumex01.in:

```
1   go atlas
2
3   mesh
4
5   x.m l=0 spac=1
6   x.m l=1 spac=1
7
8   y.m l=-0.0035 spac=0.001
9   y.m l=0 spac=0.0001
10  y.m l=0.1 spac=0.02
11  y.m l=0.5 spac=0.2
12
13  region num=1 silicon y.min=0
14  region num=2 oxide y.max=0
15
16  electrode name=gate top
17  electrode substrate
18
19  doping uniform conc=1e17 n.type
20
21  save outf=quantumex01.str
22
23  go atlas
24
25  mesh inf=quantumex01.str
26  contact name=gate n.poly
27
28  models cvt srh print
29  output con.band val.band t.quantum
30
31  method carr=2
32  solve outfile=temp.str
33  log outf=quantumex01_1.log
34  solve vgate=0 vstep=0.1 vfinal=5 name=gate ac freq=1e6 direct
35  save outf=quantumex01_1.str
36  log off
37
38  load infile=temp.str
39  solve
40  log outf=quantumex01_2.log
41  solve vgate=0 vstep=-0.2 vfinal=-2 name=gate ac freq=1e6 direct
42  save outf=quantumex01_2.str
```

```
43
44 go atlas
45
46 mesh inf=quantumex01.str
47 contact name=gate n.poly
48
49 models cvt srh print quantum
50 output con.band val.band t.quantum
51
52 method carr=2
53 solve outfile=temp.str
54 log outf=quantumex01_3.log
55 solve vgate=0 vstep=0.1 vfinal=5 name=gate ac freq=1e6 direct
56 save outf=quantumex01_3.str
57 log off
58
59 load infile=temp.str
60 solve
61 log outf=quantumex01_4.log
62 solve vgate=0 vstep=-0.2 vfinal=-2 name=gate ac freq=1e6 direct
63 save outf=quantumex01_4.str
64
65 tonyplot -overlay quantumex01_1.log quantumex01_2.log quantumex01_3.log
     quantumex01_4.log -set quantumex01_1.set
66 tonyplot -overlay quantumex01_1.str quantumex01_3.str -set
     quantumex01_2.set
67
68 extract init inf="quantumex01_1.log"
69 extract name="tox_cl" (3.9*8.85e-10*1.0e-4)/max(c."gate""substrate")
70
71 extract init inf="quantumex01_3.log"
72 extract name="tox_qm" (3.9*8.85e-10*1.0e-4)/max(c."gate""substrate")
73
74
```

14.1.2 quantumex02.in: CV Analysis Of Thin Gate Oxide NMOS Capacitor

Requires: PISCES/QUANTUM

This example demonstrates the CV characteristics of a p-type substrate MOS capacitor with a nominal 30A gate oxide. The example shows:

- Calculation of CV curves using ac Analysis
- Use of the Quantum moments quantum model for holes
- Difference between classical and quantum gate thickness
- Classical and Quantized electron density

A simple 1D MOS capacitor is created in this example with a 30A gate oxide thickness. An ac analysis is performed on the device first with the `classical` simulation and then with the `quantum` model swtiched on.

The quantum model for holes is activated by using the `p.quantum` switch in the models statement for holes. Note that this activates a quantum moments model which uses a quantum temperature calculation. This quantum temperature can be viewed in the structure files. Also, the cutline created in TONYPLOT can be used to examine the classical and quantum hole concenration differences in the structure.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

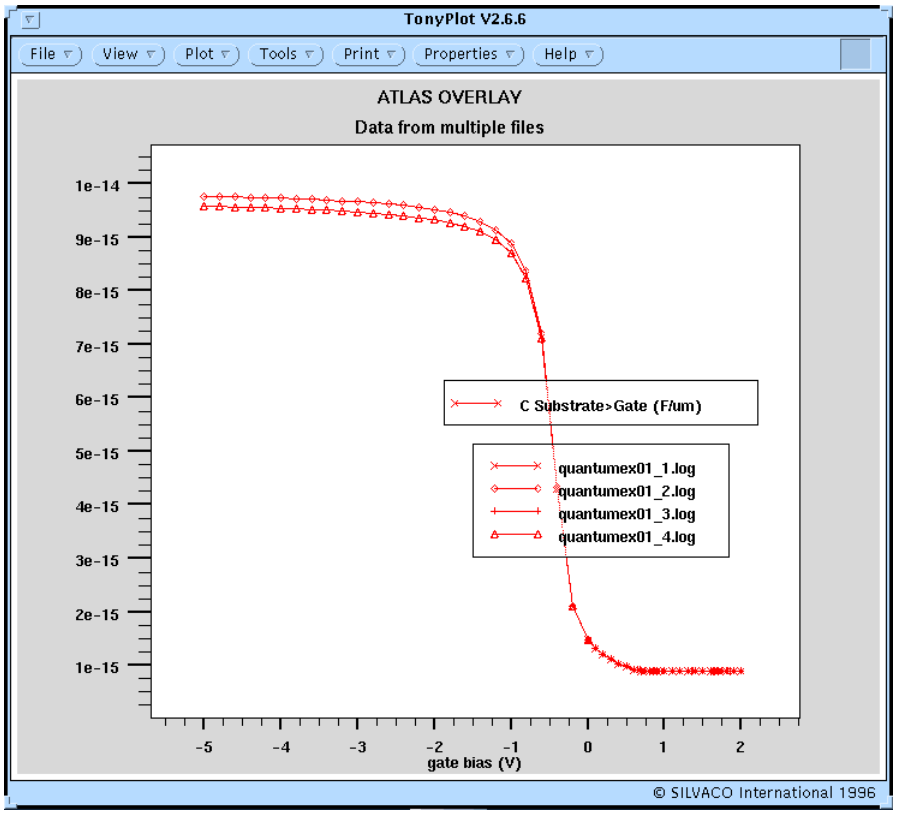


Figure 14.3: Comparison of PMOS CV curve using Quantum (plus) and Classical (x) models. An offset is seen in accumulation.

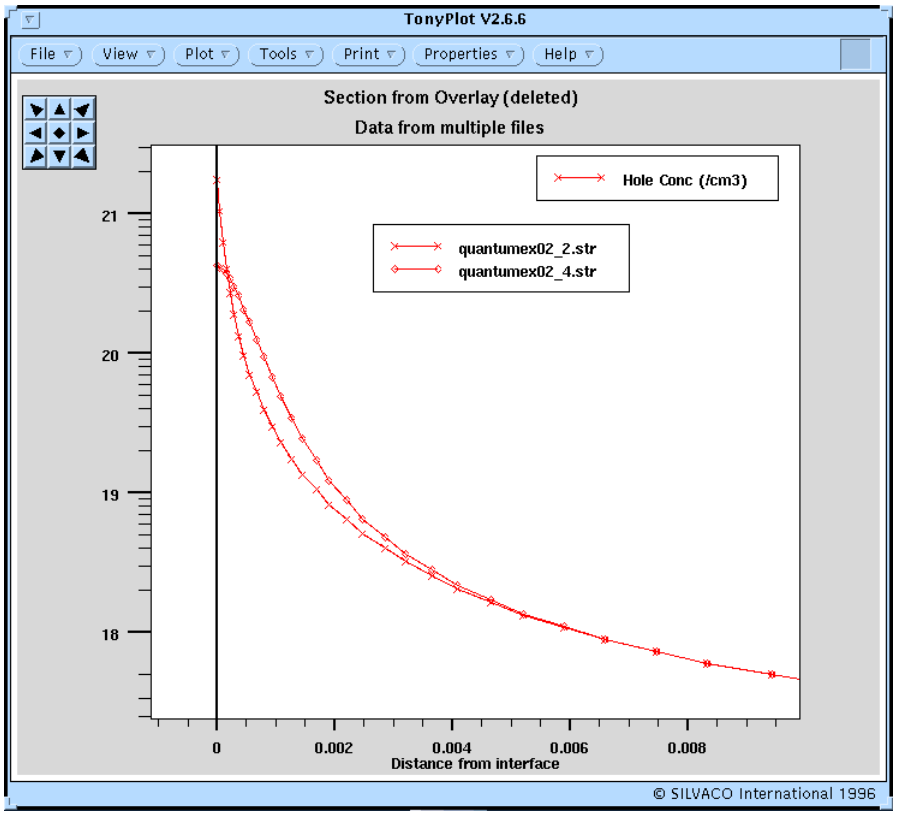


Figure 14.4: Hole concentration in the inversion layer using classical (x) and quantum (o) statistics

Input File quantum/quantumex02.in:

```
1   go atlas
2
3   mesh
4
5   x.m l=0 spac=1
6   x.m l=1 spac=1
7
8   y.m l=-0.0035 spac=0.001
9   y.m l=0 spac=0.0001
10  y.m l=0.1 spac=0.02
11  y.m l=0.5 spac=0.2
12
13  region num=1 silicon y.min=0
14  region num=2 oxide y.max=0
15
16  electrode name=gate top
17  electrode substrate
18
19  doping uniform conc=1e17 p.type
```

```
20
21 save outf=quantumex02.str
22
23 go atlas
24
25 mesh inf=quantumex02.str
26
27 models cvt srh print
28 output con.band val.band t.quantum
29
30 method carr=2 itlimit=100
31 solve outfile=temp.str
32 log outf=quantumex02_1.log
33 solve vgate=0 vstep=0.1 vfinal=2 name=gate ac freq=1e6 direct
34 save outf=quantumex02_1.str
35 log off
36
37 load infile=temp.str
38 solve
39 log outf=quantumex02_2.log
40 solve vgate=0 vstep=-0.2 vfinal=-5 name=gate ac freq=1e6 direct
41 save outf=quantumex02_2.str
42
43 go atlas
44
45 mesh inf=quantumex02.str
46
47 models cvt srh print p.quantum
48 output con.band val.band t.quantum
49
50 method carr=2
51 solve outfile=temp.str
52 log outf=quantumex02_3.log
53 solve vgate=0 vstep=0.1 vfinal=2 name=gate ac freq=1e6 direct
54 save outf=quantumex02_3.str
55 log off
56
57 load infile=temp.str
58 solve
59 log outf=quantumex02_4.log
60 solve vgate=0 vstep=-0.2 vfinal=-5 name=gate ac freq=1e6 direct
61 save outf=quantumex02_4.str
62
```

```
63 tonyplot -overlay quantumex02_1.log quantumex02_2.log quantumex02_3.log
     quantumex02_4.log -set quantumex02_1.set
64 tonyplot -overlay quantumex02_2.str quantumex02_4.str -set
     quantumex02_2.set
65
66 extract init inf="quantumex02_2.log"
67 extract name="tox_cl" (3.9*8.85e-10*1.0e-4)/max(c."gate""substrate")
68
69 extract init inf="quantumex02_4.log"
70 extract name="tox_qm" (3.9*8.85e-10*1.0e-4)/max(c."gate""substrate")
71
72
```

14.1.3 quantumex03.in: Channel Quantization in a PHEMT

Requires: BLAZE/QUANTUM

This example demonstrates the behavior of a Pseudomorphic HEMT structure using the quantum model. The device is a standard AlGaAs/GaAs/InGaAs PHEMT with a delta doped layer.

The structure is created within the ATLAS syntax. For more information on HEMT simulation see the HEMT Examples Section.

Before any external biases are applied, `qfactor` is ramped from zero to unity - this allows the inclusion of the quantum moments function to be introduced gradually into the equation set. A value of one represents a unity pre-multiplier on the quantum moments function.

With the `Quantum` model the electron concentration will typically be quite different to classical prediction. This presents some difficulty for the program in terms of the initial guess to the electron concentration. In classical simulations an excellent initial guess to the carrier concentration can be derived from the doping profile. However, with `Quantum`, this is not always the case. Thus the quantum moments effect can be turned on gradually using the parameter `qfactor`. The `qfactor` parameter might need to be ramped from very small values to unity to obtain convergence for difficult problems.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

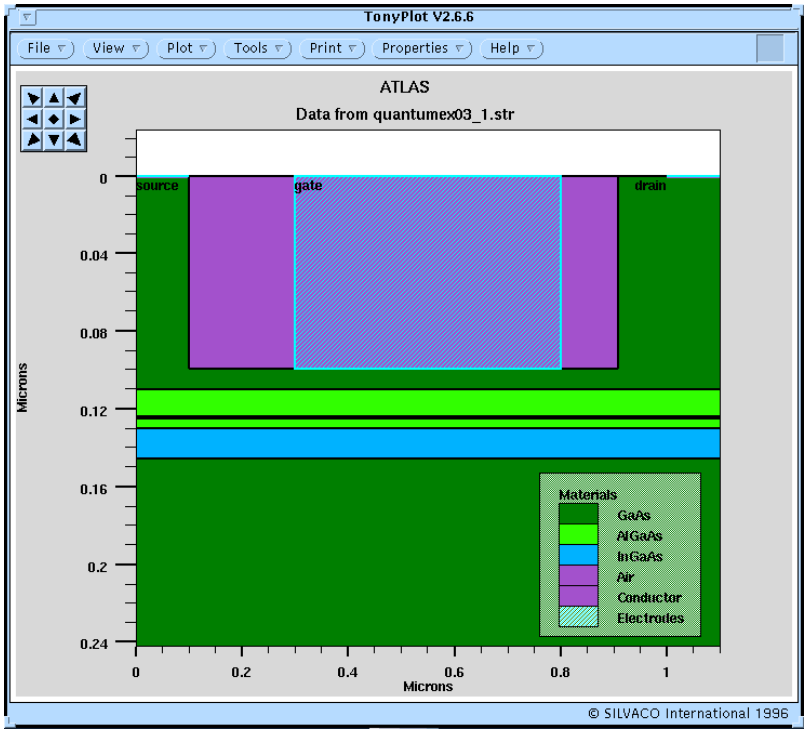


Figure 14.5: Epitaxial structure for a P-HEMT

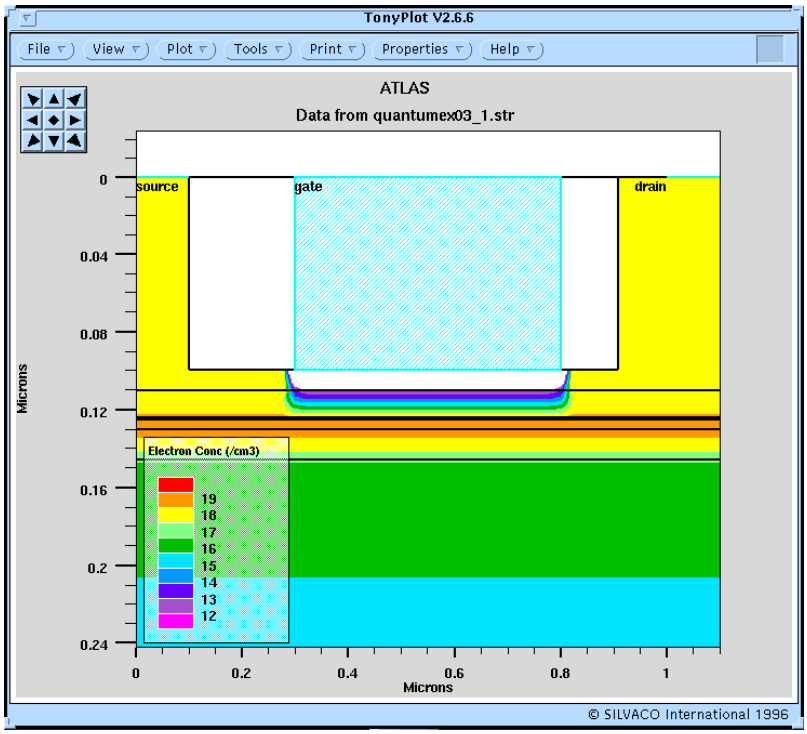


Figure 14.6: 2D electron concentration in a PHEMT using quantum models

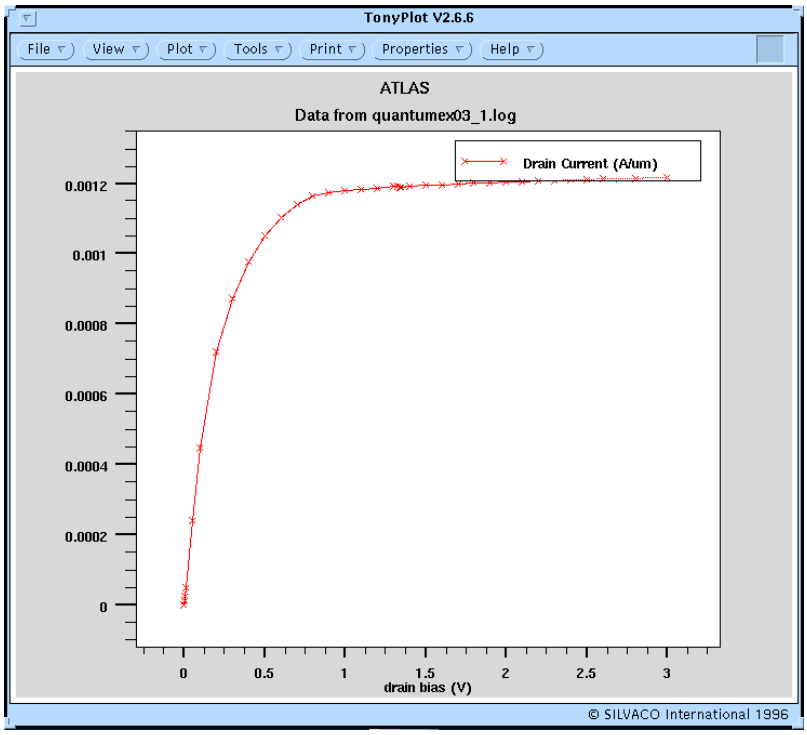


Figure 14.7: ID/VDS curve from the PHEMT using quantum models

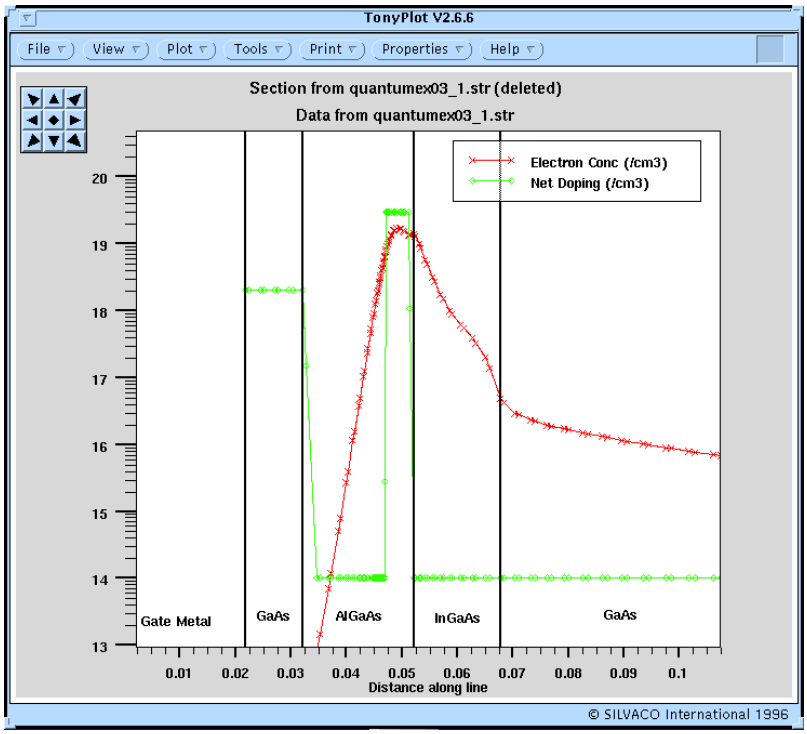


Figure 14.8: 1D section of Electron concentration in the PHEMT compared with the delta doping

Input File quantum/quantumex03.in:

```
1   go atlas
2   mesh space.mult=1.0
3   x.m l=0.0 s=0.05
4   x.m l=0.1 s=0.05
5   x.m l=0.3 s=0.02
6   x.m l=0.8 s=0.02
7   x.m l=1.0 s=0.05
8   x.m l=1.1 s=0.05
9
10  y.m l=0.0 s=0.025
11  y.m l=0.01 s=0.004
12  y.m l=0.11 s=0.0025
13  y.m l=0.124 s=0.00025
14  y.m l=0.125 s=0.00025
15  y.m l=0.130 s=0.001
16  y.m l=0.143 s=0.0025
17  y.m l=0.600 s=0.025
18
19  region num=1 material=GaAs y.max=0.01
20  region num=1 material=GaAs y.max=0.11
21  region num=2 material=AlGaAs y.min=0.11 y.max=0.124 x.comp=0.20
22  region num=3 material=AlGaAs y.min=0.124 y.max=0.125 x.comp=0.20
23  region num=4 material=AlGaAs y.min=0.125 y.max=0.130 x.comp=0.20
24  region num=5 material=InGaAs y.min=0.130 y.max=0.145 x.comp=0.85
25  region num=6 material=GaAs y.min=0.145 y.max=0.600
26  region num=7 material=air y.max=0.1 x.min=0.1 x.max=0.9
27
28  elec num=1 name=source y.min=0.0 y.max=0.0 x.min=0.0 x.max=0.1
29  elec num=2 name=drain  y.min=0.0 y.max=0.0 x.min=1.0 x.max=1.1
30  elec num=3 name=gate   y.min=0.0 y.max=0.1 x.min=0.3 x.max=0.80
31
32  doping uniform n.type conc=1e14
33  doping uniform n.type region=4 conc=3e19
34  doping uniform n.type region=1 conc=2.0e18
35
36  contact name=gate workf=3.3
37  material align=0.3
38  model  numcarr=1 fldmob conmob srh print quantum
39
40  output con.band val.band band.param t.quantum
41  method maxtrap=8
42  solve qfactor=0.0
```

```
43 solve qfactor=0.0001
44 solve qfactor=0.001
45 solve qfactor=0.01
46 solve qfactor=0.1
47 solve qfactor=1.0
48
49
50 log outf=quantumex03_1.log
51 solve
52 save outf=quantumex03_1.str master
53
54 tonyplot quantumex03_1.str  -set quantumex03_1.set
55
56 solve vdrain=0.01
57 solve vdrain=0.05
58 solve vdrain=0.1 vstep=0.1 vfinal=3.0 name=drain
59
60 tonyplot quantumex03_1.log  -set quantumex03_2.set
61
62 quit
63
```

14.1.4 quantumex04.in: Heterojunction Diode Leakage

Requires: BLAZE/QUANTUM

This example demonstrates the difference between an AlGaAs/GaAs heterojunction diode modeled with and without quantum physics.

In the first half of the file, the example calculates the forward and reverse characteristics of a classically modeled diode. In the second half of the example, the quantum model is switched on. Note that, as in the previous example, the `qfactor` is used to ramp the level of quantum moments included in the example for zero to unity.

The overlayed results show around an order of magnitude higher reverse leakage in the quantum case. This is due to the{\italic} smearing of the electron concentration accross the heterojunction.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

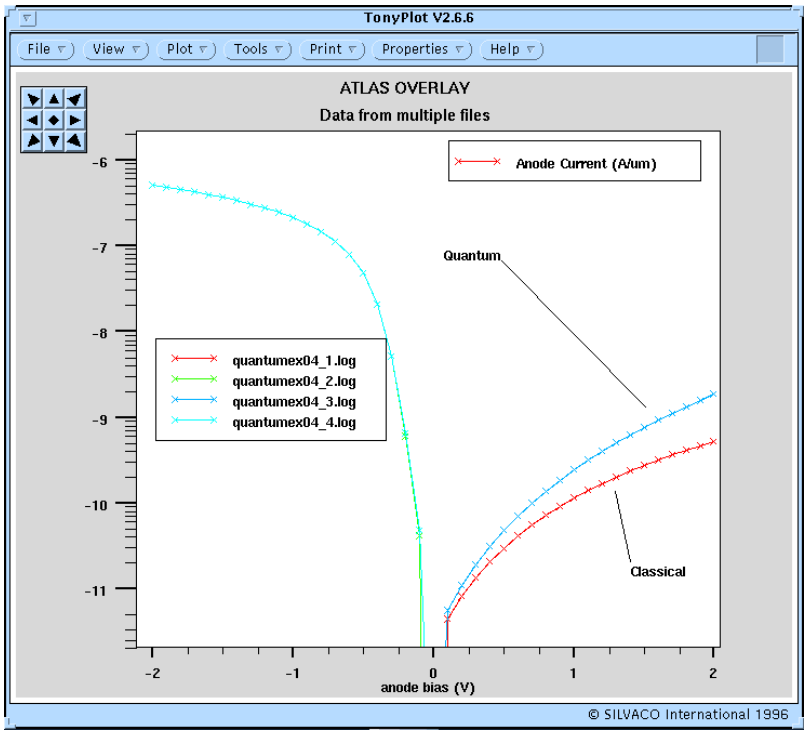


Figure 14.9: Increased leakage in a reverse biased heterojunction diode due to quantum statistics as compared to classical

Input File quantum/quantumex04.in :

```
go atlas

mesh

x.m l=0 spac=1
x.m l=1 spac=1

y.m l=0.000 spac=0.1
y.m l=0.5 spac=0.0001
y.m l=1.0 spac=0.1

region num=1 material=GaAs
region num=2 material=AlGaAs y.min=0.000 y.max=0.4995 x.comp=0.8
  grad.34=0.0005

electrode name=anode top
electrode name=cathode bottom

doping uniform conc=1e14 n.type

models fldmob srh print
```

```
21
22 material material=AlGaAs align=0.6
23 output band.param val.band con.band
24
25 method carr=1 elec  newton
26
27 solve vanode=0.00 outf=temp
28
29 log outf=quantumex04_1.log
30 solve vanode=0 vstep=0.1 vfinal=2 name=anode
31
32 log outf=quantumex04_2.log
33 load inf=temp
34 solve vanode=0 vstep=-0.1 vfinal=-2 name=anode
35 save outf=quantumex04_1.str
36
37
38 go atlas
39
40 mesh
41
42 x.m l=0 spac=1
43 x.m l=1 spac=1
44
45 y.m l=0.000 spac=0.1
46 y.m l=0.5 spac=0.0001
47 y.m l=1.0 spac=0.1
48
49 region num=1 material=GaAs
50 region num=2 material=AlGaAs y.min=0.000 y.max=0.4995 x.comp=0.8
     grad.34=0.0005
51
52 electrode name=anode top
53 electrode name=cathode bottom
54
55 doping uniform conc=1e14 n.type
56
57 models fldmob srh print quantum
58
59 material material=AlGaAs align=0.6
60 output t.quantum band.param val.band con.band
61
62 method carr=1 elec  newton
```

```
63 solve qfactor=0
64 solve qfactor=0.01
65 solve qfactor=0.05
66 solve qfactor=0.1
67 solve qfactor=0.2
68 solve qfactor=0.3
69 solve qfactor=0.4
70 solve qfactor=0.5
71 solve qfactor=0.6
72 solve qfactor=0.7
73 solve qfactor=1.0 outfile=temp
74
75
76 log outf=quantumex04_3.log
77 solve vanode=0 vstep=0.1 vfinal=2 name=anode
78
79 log outf=quantumex04_4.log
80 load inf=temp
81 solve vanode=0 vstep=-0.1 vfinal=-2 name=anode
82 save outf=quantumex04_2.str
83
84 tonyplot -overlay quantumex04_1.log quantumex04_2.log quantumex04_3.log
     quantumex04_4.log -set quantumex04_1.set
85
```

This page is intentionally left blank

15.1. TFT: TFT Application Examples

15.1.1 tftex01.in: Amorphous Silicon TFT: Passivated Device

Requires: SPISCES/TFT

This file performs Id/Vgs simulation of a TFT device with material properties corresponding to passivated alpha-Si:H material. The example shows:

- structure formation using ATLAS syntax
- material and model settings for passivated a-Si
- forward Id/Vgs characteristics

The key command in TFT simulation is the `defect` statement. It is used to define a continuous density of trap states in the silicon and the relevant trapping cross-sections.

The Id/Vgs ramping is done in a similar manner to the threshold voltage tests for MOS devices described in the MOS example. Results from this example can be compared with the un-passivated a-Si device.

A more detailed description of TFT material settings is given in the *Forward/Reverse Gate Voltage Characteristic* example.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

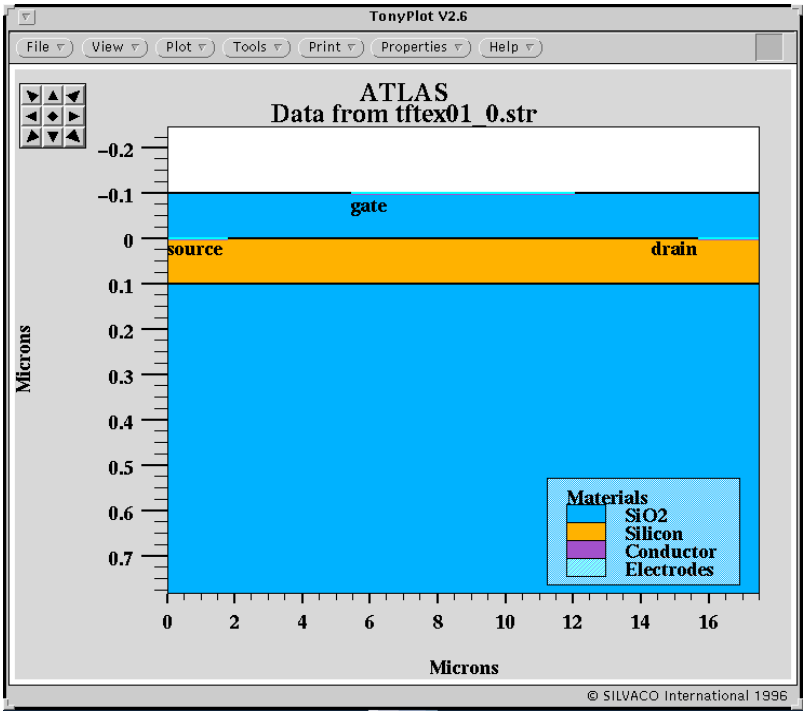


Figure 15.1: Layer structure of a Planar a-Si TFT defined using ATLAS syntax

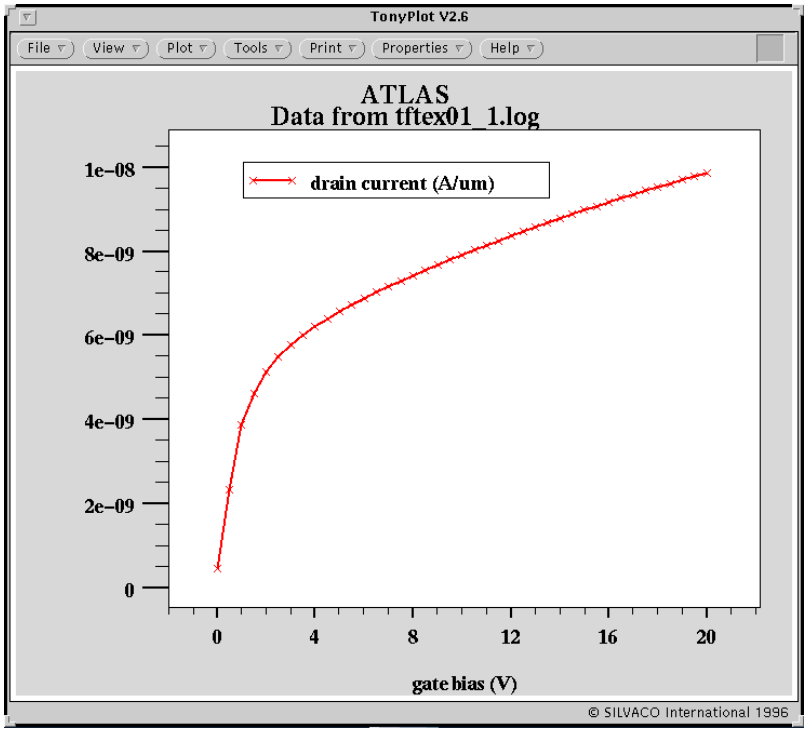


Figure 15.2: Id/Vgs for the a-Si device with passivated DOS defined in the DEFECT statement

Input File tft/tftex01.in :

```
1   go atlas
2   TITLE   a-si TFT simulation
3   # SILVACO International 1993, 1994
4
5   mesh          nx=30  ny=30
6   #
7   x.m          n=1      l=0          r=1
8   x.m          n=30     l=17.5       r=1
9   #
10  #
11  y.m          n=1      l=-0.1    r=1
12  y.m          n=5      l=0       r=1
13  y.m          n=20     l=0.1     r=1
14  y.m          n=30     l=10.     r=1.6
15  #
16  #
17  #  *************** regions ***************
18  #             1=oxide  2=silicon 3-oxide
19  #
20  region       num=1  y.max=0.    oxide
21  region       num=2  y.min=0.    y.max=0.1   silicon
```

```
region       num=3  y.min=0.1   oxide
#
#  *************** electrodes ***************
#    1=gate  2=substrate  3=source  4=drain
#
elec  num=1  x.min=5    x.max=12.5  y.min=-0.1 y.max=-0.1 name=gate
elec  num=2  substrate              name=substrate
elec  num=3  x.min=0.   x.max=2.    y.min=0.   y.max=0.   name=source
elec  num=4  x.min=15.5 x.max=17.5  y.min=0.   y.max=0.   name=drain
#
#  *************** doping profiles  ***************

doping     reg=2  uniform conc=7.e14 n.type
doping     reg=2  gauss conc=3.e18 n.type x.right=5 char=0.3
doping     reg=2  gauss conc=3.e18 n.type x.left=15 char=0.3


#   Set parameters for amorphus silicon
#
material region=2 mun=20 mup=1.5 nc300=2.5e20 \
 nv300=2.5e20 eg300=1.9
#
defects nta=1.e21 ntd=1.e21 wta=0.033 wtd=0.049 \
  nga=1.5e15 ngd=1.5e15 ega=0.62 egd=0.78 wga=0.15 wgd=0.15 \
  sigtae=1.e-17 sigtah=1.e-15 sigtde=1.e-15 sigtdh=1.e-17  \
  siggae=2.e-16 siggah=2.e-15 siggde=2.e-15 siggdh=2.e-16
#
contactnum=1 alum
models  temp=300
#

method itlimit=30
solve init

save outf=tftex01_0.str

tonyplot  tftex01_0.str -set tftex01_0.set

#
# Electrode #1 - Gate, #2 - Substrate, #3 - Source, #4 - Drain
#

method  newton
```

```
65 solve  vdrain=0.1
66 solve  vdrain=0.2
67 solve  vdrain=0.5
68 solve  vdrain=1 vfinal=5 vstep=1 name=drain
69 #
70 # Solve id versus vg curve
71 #
72
73 #
74 log outf=tftex01_1.log
75 solve vstep=0.5 vfinal=20. name=gate vgate=0. vdrain=5
76 #
77 tonyplot  tftex01_1.log  -set tftex01_log.set
78
79 quit
80
```

15.1.2 tftex02.in: Amorphous Silicon TFT: Un-Passivated Device

Requires: S-PISCES/TFT

This file performs Id/Vgs simulation of a TFT device with material properties corresponding to un-passivated alpha-Si:H material. The example shows:

- structure formation using ATLAS syntax
- material and model settings for un-passivated a-Si
- forward Id/Vgs characteristics

The key command in TFT simulation is the `defect` statement. It is used to define a continuous density of trap states in the silicon and the relevant trapping cross-sections.

The Id/Vgs ramping is done in a similar manner to the threshold voltage tests for MOS devices described in the MOS example. Results from this example can be compared with the passivated a-Si device.

A more detailed description of TFT material settings is given in the *Forward/Reverse Gate Voltage Characteristic* example.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

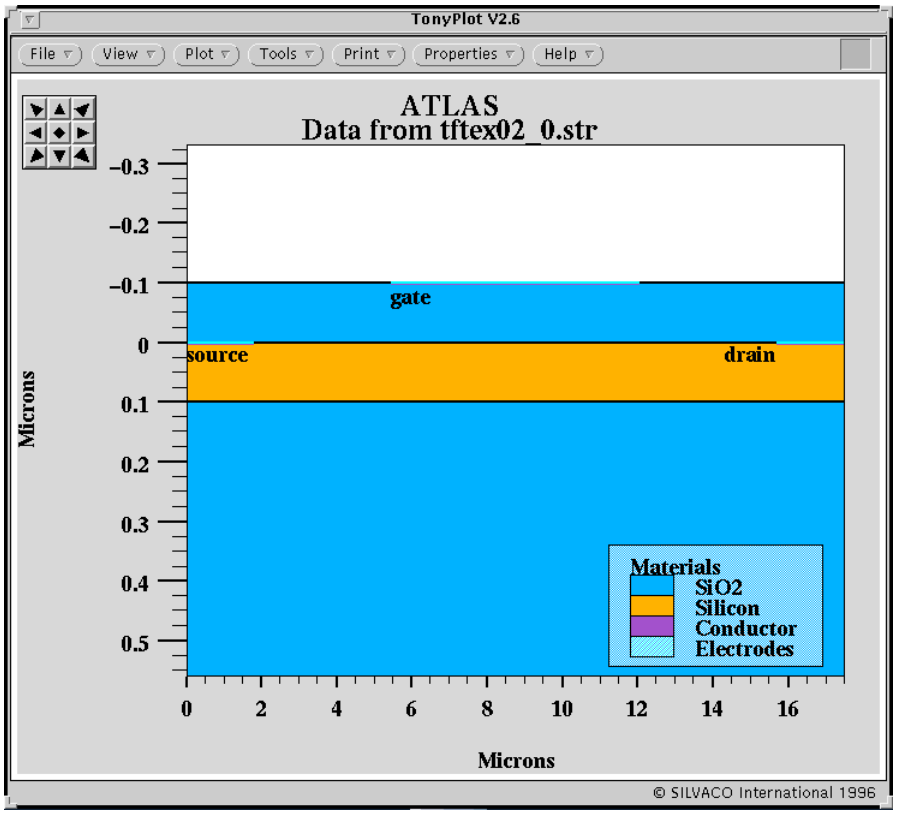


Figure 15.3: Layer structure of a Planar a-Si TFT defined using ATLAS syntax

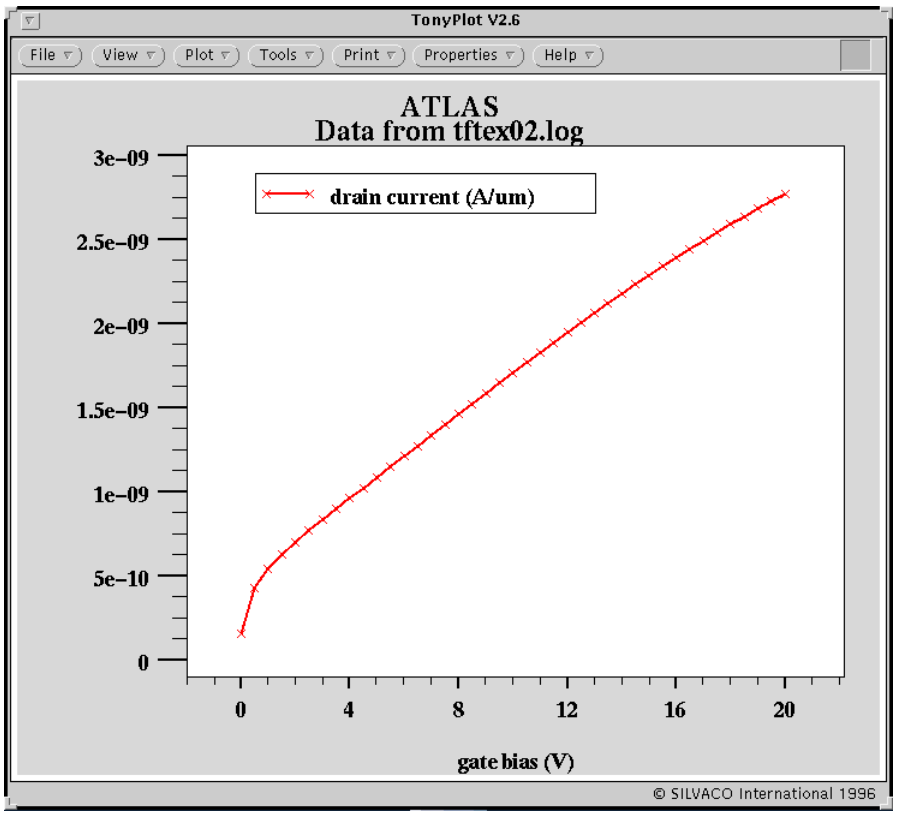


Figure 15.4: Id/Vgs for the a-Si device with un-passivated DOS defined in the DEFECT statement

Input File tft/tftex02.in :

```
1   go atlas
2   TITLE   a-si TFT simulation
3   # SILVACO International 1993, 1994
4   #
5   mesh         nx=30  ny=30
6   #
7   x.m          n=1      l=0         r=1
8   x.m          n=30     l=17.5      r=1
9   #
10  #
11  y.m          n=1      l=-0.1    r=1
12  y.m          n=5      l=0       r=1
13  y.m          n=20     l=0.1     r=1
14  y.m          n=30     l=10.     r=1.6
15  #
16  #
17  #  ***************** regions ***************
18  #             1=oxide  2=silicon 3-oxide
19  #
```

```
region       num=1  y.max=0.    oxide
region       num=2  y.min=0.    y.max=0.1   silicon
region       num=3  y.min=0.1   oxide
#
#  *************** electrodes ***************
#    1=gate  2=substrate  3=source  4=drain
#
elec  num=1  x.min=5    x.max=12.5  y.min=-0.1 y.max=-0.1 name=gate
elec  num=2  substrate              name=substrate
elec  num=3  x.min=0.   x.max=2.    y.min=0.   y.max=0.   name=source
elec  num=4  x.min=15.5 x.max=17.5  y.min=0.   y.max=0.   name=drain
#
#  *************** doping profiles  ***************

doping     reg=2  uniform conc=7.e14 n.type
doping     reg=2  gauss conc=3.e18 n.type x.right=5 char=0.3
doping     reg=2  gauss conc=3.e18 n.type x.left=15 char=0.3

material region=2 mun=20 mup=1.5 nc300=2.5e20 \
  nv300=2.5e20 eg300=1.9

#
defects nta=1.e21 ntd=1.e21 wta=0.033 wtd=0.049 \
  nga=4.5e15 ngd=4.5e15 ega=0.62 egd=0.78 wga=0.15 wgd=0.15 \
  sigtae=1.e-17 sigtah=1.e-15 sigtde=1.e-15 sigtdh=1.e-17 \
  siggae=2.e-16 siggah=2.e-15 siggde=2.e-15 siggdh=2.e-16
#
contactnum=1 alum
models  temp=300
#


solve init

save outf=tftex02_0.str
tonyplot tftex02_0.str -set tftex02_0.set

#
# Electrode #1 - Gate, #2 - Substrate, #3 - Source, #4 - Drain
#

method newton

```

```
63 solve  vdrain=0.1
64 solve  vdrain=0.2
65 solve  vdrain=0.5
66 solve  vdrain=1 vfinal=5 vstep=1 name=drain
67 #
68 # Solve id versus vg curve
69 #
70
71 #
72 log outf=tftex02.log
73 solve vstep=0.5 vfinal=20. name=gate vgate=0. vdrain=5
74 #
75 tonyplot  tftex02.log  -set tftex02_log.set
76 quit
77
```

15.1.3 tftex03.in: Polysilicon TFT : Passivated Device

Requires: S-PISCES/TFT

This file performs Id/Vgs simulation of a TFT device with material properties corresponding to passivated polysilicon material. The example shows:

- structure formation using ATLAS syntax
- material and model settings for passivated polysilicon
- forward Id/Vgs characteristics

The key command in TFT simulation is the `defect` statement. It is used to define a continuous density of trap states in the silicon and the relevant trapping cross-sections.

The Id/Vgs ramping is done in a similar manner to the threshold voltage tests for MOS devices described in the MOS example. Results from this example can be compared with the un-passivated polysilicon device.

A more detailed description of TFT material settings is given in the *Forward/Reverse Gate Voltage Characteristic* example.

This corresponds to a passivated polysilicon material. To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

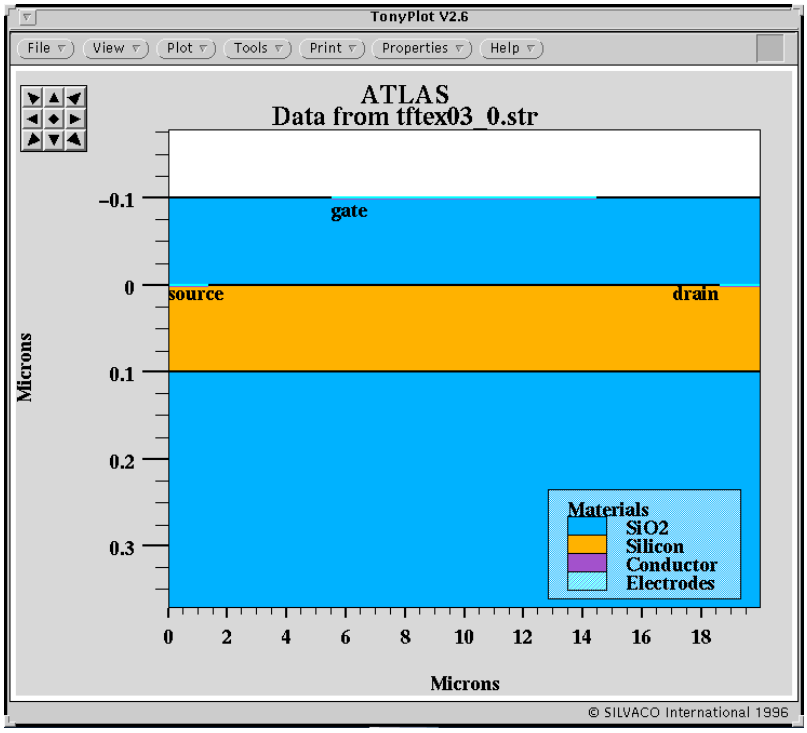


Figure 15.5: Layer structure of a polysilicon TFT defined using ATLAS syntax (zoom)

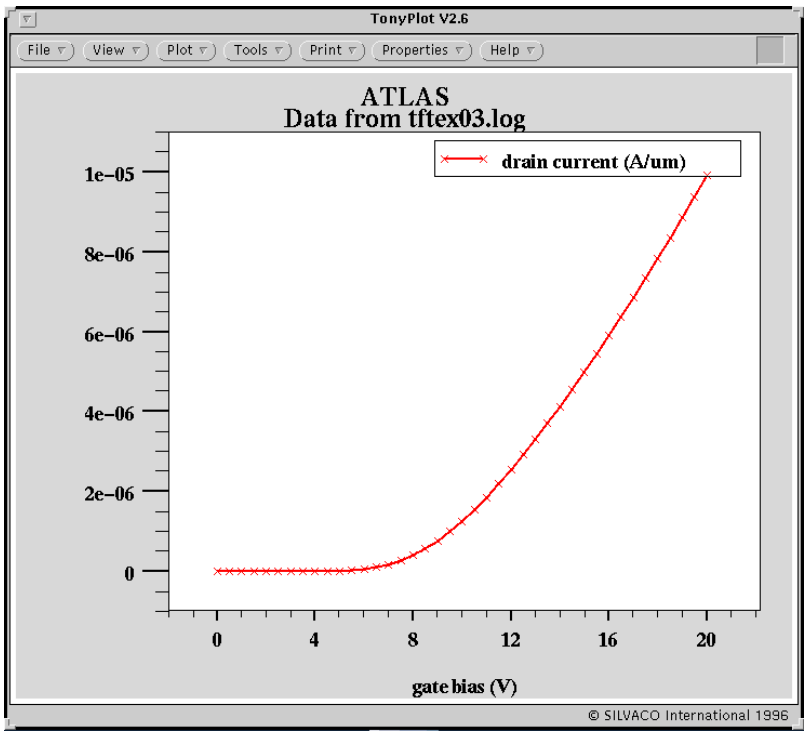


Figure 15.6: Id/Vgs for the polysilicon device with passivated DOS defined in the DEFECT statement

Input File tft/tftex03.in:

```
1   go atlas
2   TITLE polysilicon TFT simulation
3   # SILVACO International 1993, 1994
4
5   mesh   nx=30  ny=30
6   #
7   x.m          n=1      l=0         r=1
8   x.m          n=30     l=20.       r=1
9   #
10  #
11  y.m          n=1      l=-0.1    r=1
12  y.m          n=5      l=0       r=1
13  y.m          n=19     l=0.1     r=1
14  y.m          n=30     l=10.     r=1.6
15  #
16  #
17  #  *************** regions ******************************
18  #             1=oxide  2=silicon 3-oxide
19  #
20  region       num=1  y.max=0.    oxide
21  region       num=2  y.min=0. y.max=0.1   silicon
22  region       num=3  y.min=0.1   oxide
23  #
24  #  *************** electrodes ***************************
25  #    1=gate  2=substrate  3=source  4=drain
26  #
27  elec  num=1  x.min=5   x.max=15  y.min=-0.1 y.max=-0.1 name=gate
28  elec  num=2  substrate            name=substrate
29  elec  num=3  x.min=0.  x.max=2.  y.min=0.   y.max=0.   name=source
30  elec  num=4  x.min=18. x.max=20. y.min=0.   y.max=0.   name=drain
31  #
32  #  *************** doping profiles **********************
33  #
34  doping     reg=2  uniform conc=1.e11 n.type
35  doping     reg=2  gauss conc=1.e20 n.type x.right=5 char=0.3
36  doping     reg=2  gauss conc=1.e20 n.type x.left=15 char=0.3
37
38  #
39  #
40  #     Set parameters for polysilicon
41  #
42  material region=2 mun=300 mup=30
```

```
#
defects nta=1.12e21 ntd=4.e20 wta=0.025 wtd=0.05 \
  nga=5.e17 ngd=1.5e18 ega=0.4 egd=0.4 wga=0.1 wgd=0.1 \
  sigtae=1.e-16 sigtah=1.e-14 sigtde=1.e-14 sigtdh=1.e-16  \
  siggae=1.e-16 siggah=1.e-14 siggde=1.e-14 siggdh=1.e-16
#
contactnum=1 n.polysilicon
models  temp=300
#


solve init
save outf=tftex03_0.str
tonyplot tftex03_0.str -set tftex03_0.set


#
# Electrode #1 - Gate, #2 - Substrate, #3 - Source, #4 - Drain
#

method newton
#
solve  vdrain=0.1
solve  vdrain=0.2
solve  vdrain=0.5
solve  vdrain=1 vstep=1. vfinal=5. name=drain
#
# Solve id versus vg curve
#

#
log outf=tftex03.log
solve vstep=0.5 vfinal=20 name=gate vgate=0. vdrain=5

tonyplot  tftex03.log -set tftex03_log.set
#
quit

```

15.1.4 tftex04.in: Polysilicon TFT : Un-Passivated Device

Requires: S-PISCES/TFT

This file performs Id/Vgs simulation of a TFT device with material properties corresponding to un-passivated polysilicon material. The example shows:

- structure formation using ATLAS syntax
- material and model settings for passivated polysilicon
- forward Id/Vgs characteristics

The key command in TFT simulation is the `defect` statement. It is used to define a continuous density of trap states in the silicon and the relevant trapping cross-sections.

The Id/Vgs ramping is done in a similar manner to the threshold voltage tests for MOS devices described in the MOS example. Results from this example can be compared with the passivated polysilicon device.

A more detailed description of TFT material settings is given in the *Forward/Reverse Gate Voltage Characteristic* example.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

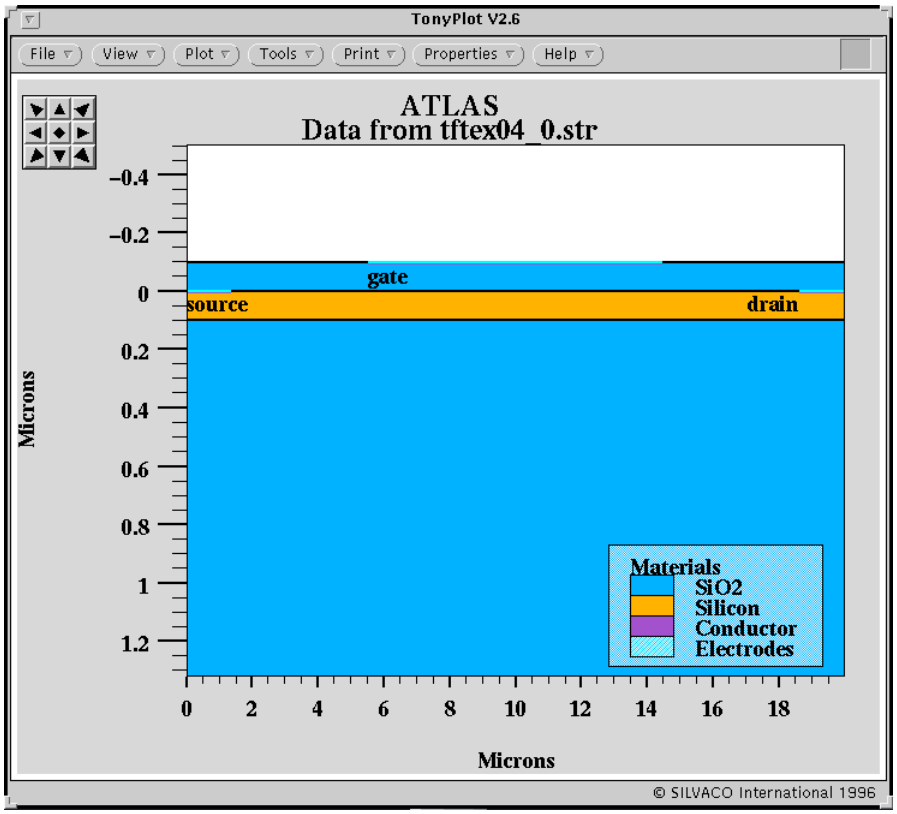


Figure 15.7: Layer structure of a polysilicon TFT defined using ATLAS syntax

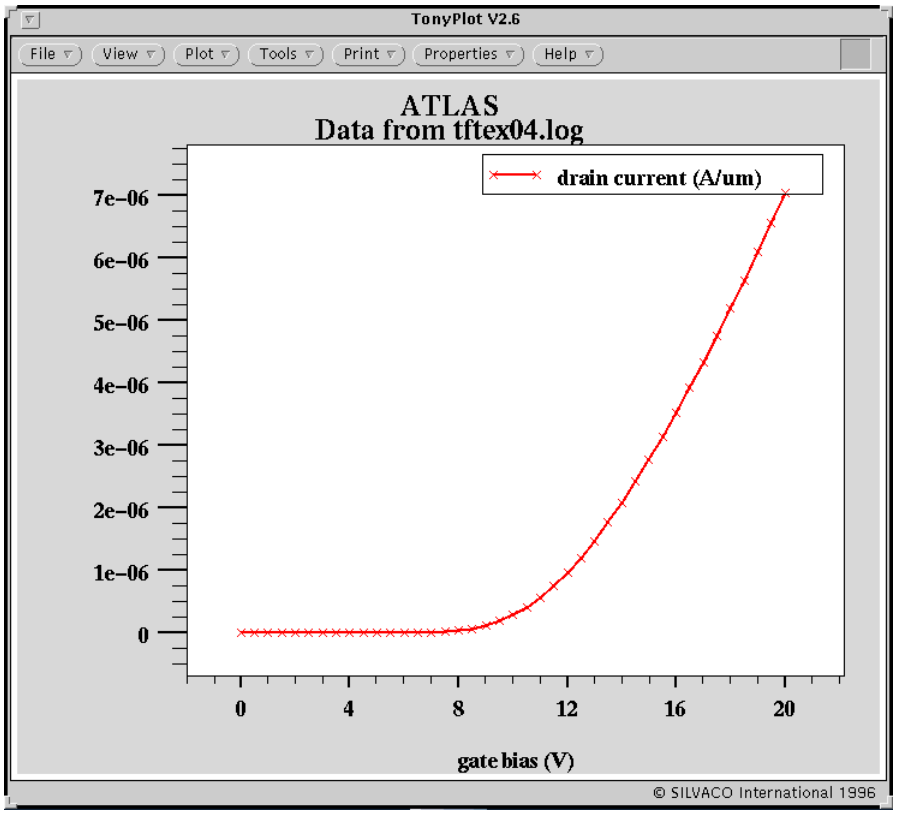


Figure 15.8: Id/Vgs for the polysilicon device with un-passivated DOS defined in the DEFECT statement

Input File tft/tftex04.in:

```
1   go atlas
2   TITLE polysilicon TFT simulation
3   # SILVACO International 1993, 1994
4
5   mesh  nx=30  ny=30
6   #
7   x.m          n=1      l=0          r=1
8   x.m          n=30     l=20.        r=1
9   #
10  #
11  y.m          n=1      l=-0.1    r=1
12  y.m          n=5      l=0       r=1
13  y.m          n=19     l=0.1     r=1
14  y.m          n=30     l=10.     r=1.6
15  #
16  #
17  #  ***************** regions *******************************
18  #             1=oxide  2=silicon 3-oxide
19  #
```

```
region       num=1  y.max=0.    oxide
region       num=2  y.min=0. y.max=0.1   silicon
region       num=3  y.min=0.1   oxide
#
#  *************** electrodes ***************************
#    1=gate  2=substrate  3=source  4=drain
#
elec  num=1  x.min=5   x.max=15  y.min=-0.1 y.max=-0.1 name=gate
elec  num=2  substrate            name=substrate
elec  num=3  x.min=0.  x.max=2.  y.min=0.   y.max=0.   name=source
elec  num=4  x.min=18. x.max=20. y.min=0.   y.max=0.   name=drain
#
#  *************** doping profiles ***********************
#
doping     reg=2  uniform conc=1.e11 n.type
doping     reg=2  gauss conc=1.e20 n.type x.right=5 char=0.3
doping     reg=2  gauss conc=1.e20 n.type x.left=15 char=0.3

#
#
#     Set parameters for polysilicon
#
material region=2 mun=300 mup=30
#
defects nta=1.12e21 ntd=4.e20 wta=0.025 wtd=0.05 \
  nga=1.e18 ngd=3.e18 ega=0.4 egd=0.4 wga=0.1 wgd=0.1 \
  sigtae=1.e-16 sigtah=1.e-14 sigtde=1.e-14 sigtdh=1.e-16  \
  siggae=1.e-16 siggah=1.e-14 siggde=1.e-14 siggdh=1.e-16
#
contactnum=1 n.polysilicon
models  temp=300
#

method
solve init
save outf=tftex04_0.str
tonyplot  tftex04_0.str -set tftex04_0.set


#
# Electrode #1 - Gate, #2 - Substrate, #3 - Source, #4 - Drain
#

```

```
63  method  newton
64
65  solve  vdrain=0.1
66  solve  vdrain=0.2
67  solve  vdrain=0.5
68  solve  vdrain=1 vstep=1. vfinal=5. name=drain
69
70  #
71  # Solve id versus vg curve
72  #
73
74  #
75  log outf=tftex04.log master.out
76  solve vstep=0.5 vfinal=20 name=gate v1=0. vdrain=5
77  #
78  tonyplot tftex04.log -set tftex04_log.set
79  quit
80
```

15.1.5 tftex05.in: Forward/Reverse Gate Voltage Characteristic

Requires: SSUPREM4/S-PISCES/TFT

This example demonstrates how integrated process and device TFT simulations can be performed in ATHENA and ATLAS. The input files consists of :

- construction of TFT device in ATHENA
- forward (positive) Vgs sweep in ATLAS
- reverse (negative) Vgs sweep in ATLAS

The first part of the file uses ATHENA to construct the geometry and doping of a TFT device. The starting substrate is defined as silicon dioxide to emulate the flat panel display glass. The transistor is simulated with a metal gate on the bottom and a gate insulator made from oxide and nitride. A lightly doped silicon layer is deposited to act as the channel region. A heavily doped layer is placed on top to become the source/drain regions. Single crystal silicon or polysilicon could be used equally well at this stage. However the important electrical properties of the material are set in ATLAS, so at this stage it is not important which material is used. Metal for the source/drain contacts is then applied. Then an etchback through the metal and heavily doped silicon is done. Some of the lightly doped silicon layer is also removed. This final etch separates the source and drain. The final stage in ATHENA is to define the electrodes for use in ATLAS.

The ATLAS part of this file is used to simulate the gate voltage bias from -20V to +20V with the drain at +10V. Before the biasing is applied it is first necessary to set all the relevant material parameters.

In simulating TFTs, the most important requirement is the setting of the `defect` statement, specifying the density of states in the semiconductor bandgap. The defect states are specified as donor-like and acceptor-like and as tail and mid-gap gaussian states. This gives a total of four distributions each with their own trapping cross sections for electrons and holes for a total of eight cross section values. The values included in this example are typical of amorphous silicon used in TFT

transistors, but each user has a different process. The defect densities are sensitive to the hydrogen annealing of the device and so must be tuned by the user.

The `material` statement is used to set the material constants of the semiconductor to those of amorphous silicon. The interface step defines an interface charge on each semiconductor/insulator interface. It is possible to vary this charge by position using the bounding box parameters on the interface statement.

The models needed for TFT simulation are simple. A constant mobility as defined by the material statement is used and SRH recombination is included. In order to simulate reverse leakage the band-band tunneling model is included by: `models bbt.std.` The exponential parameter of the band-band tunneling model is adjusted for amorphous silicon.

The first step in the solve sequence is to ramp the drain voltage up to+10V. At this point a file is stored. Then the reverse gate voltage ramp is applied. The gate voltage is ramped up to -20V. The drain current will increase during this ramp due to tunneling current in the TFT. The magnitude of this is controlled by the parameters of the band-band tunneling model. at the end of the ramp the final drain current is extracted.

At the highest reverse gate bias, a structure file is saved. By plotting the recombination in the device at this bias it is possible to see the band-band generation current as a negative recombination term.

The final step is the forward gate voltage sweep. This is done by first loading the solution saved with zero gate voltage. Then the gate is ramped to +20V. From this data the `extract` syntax is used to get the sub-threshold leakage slope. This slope, which is typically 0.5-1.5 V/decade is sensitive to the defect distribution in the semiconductor.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

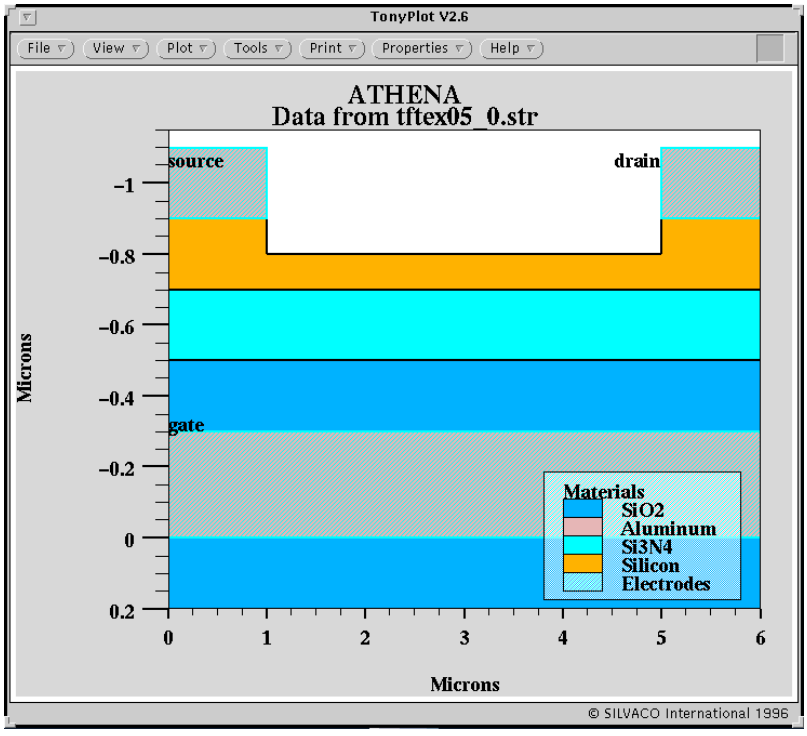


Figure 15.9: Non-planar TFT simulated in ATHENA. The gate is beneath the channel.

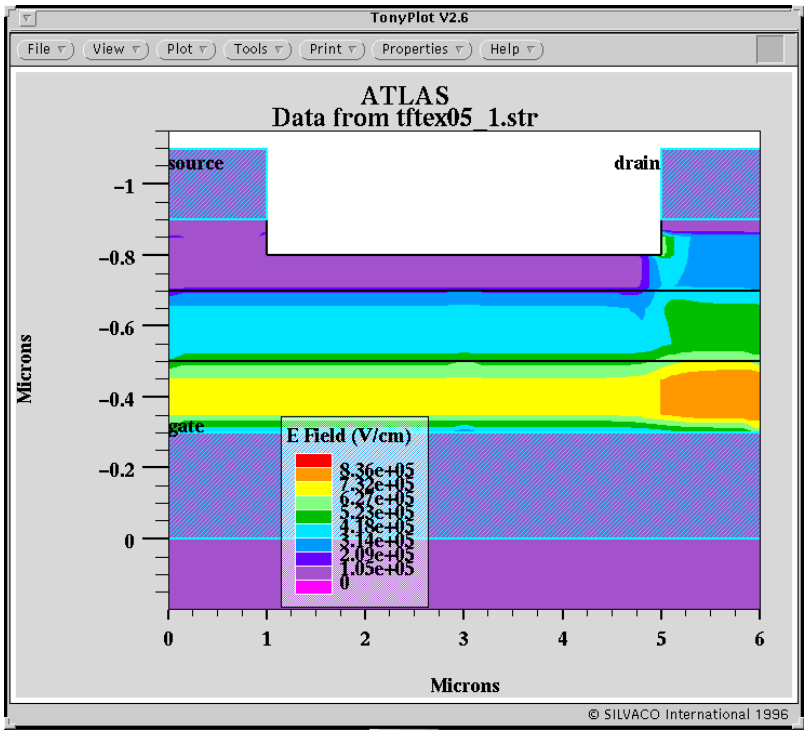


Figure 15.10: Electric field in the TFT during forward gate biasing

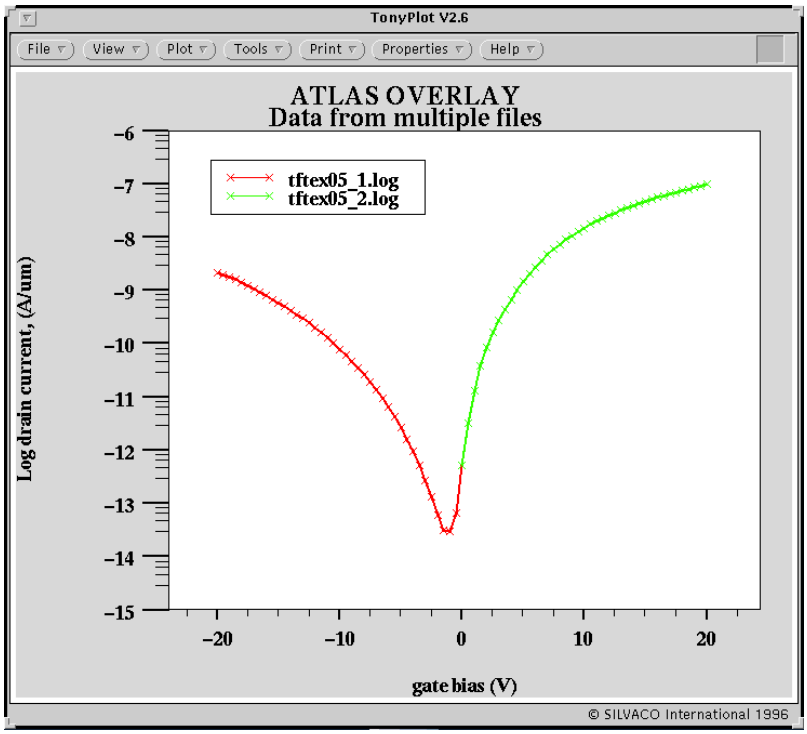


Figure 15.11: Forward and Reverse Id/Vgs curve showing strong reverse leakage due to tunneling generation

Input File tft/tftex05.in:

```
1   go athena
2   # ATHENA TFT formation -> ATLAS Id/Vg extraction example
3   # Silvaco International 1994
4
5   # mesh definition
6   line x loc=0   spac=0.2
7   line x loc=1.0 spac=0.1
8   line x loc=3.0 spac=0.25
9
10  line y loc=0   spac=0.1
11  line y loc=0.2 spac=0.1
12
13  # start with glass substrate
14  initialize oxide
15
16  # deposit thin metal as gate
17  deposit  aluminum  thick=0.30   spaces=2
18
19  # MIS insulator
20  deposit  oxide     thick=0.20   spaces=4
21  deposit  nitride   thick=0.2    spaces=4
22
23  # active amorphous silicon layer
24  deposit  silicon   thick=0.15   div=10 c.phos=1e14
25  deposit  silicon   thick=0.05   c.phos=1.0e20  div=4
26
27  # deposit source/drian contacts
28  deposit  aluminum    thick=0.2    div=2
29
30  # pattern source/drain regions. Recess into active layer.
31  deposit  barrier  thick=0.1
32  etch     barrier  p1.x=1.0 right
33  etch     aluminum  dry thick=0.20
34  etch     silicon   dry thick=0.10
35  etch     barrier   all
36
37  # mirror to get full structure
38  structure mirror right
39
40  # define electrodes for ATLAS
41  electrode name=source x=0 y=-1.0
42  electrode name=gate   x=3
```

```
43 electrode name=drain  x=6 y=-1.0
44
45 structure out=tftex05_0.str
46
47 tonyplot  tftex05_0.str -set tftex05_0.set
48
49
50 go atlas
51 # Title   a-si TFT simulation
52 # Silvaco International 1993, 1994
53
54 #   set parameters for amorphous silicon
55
56 material material=silicon mun=20 mup=1.5 nc300=2.5e20 \
57  nv300=2.5e20 eg300=1.8 taun0=1e-8 taup0=1e-8
58
59 defects nta=1.e21 ntd=1.e21 wta=0.05 wtd=0.05 \
60   nga=1.0e16 ngd=1.0e16 ega=0.6 egd=0.6 wga=0.3 wgd=0.3 \
61   sigtae=1.e-17 sigtah=1.e-15 sigtde=1.e-15 sigtdh=1.e-17  \
62   siggae=2.e-16 siggah=2.e-15 siggde=2.e-15 siggdh=2.e-16
63 interface qf=3e10
64
65 # define gate workfunction
66 contactname=gate alum
67
68 # define models, include bbt.kl for reverse leakage effect
69 models srh temp=300 print bbt.kl bb.b=7.5e6
70
71 # first ramp the drain up
72
73 method newton
74 solve init
75 solve prev vdrain=0.1
76 solve prev vdrain=0.2
77 solve prev vdrain=0.5
78 solve vdrain=1 vfinal=10 vstep=1 name=drain
79
80 # select to store total electric field and save a file
81 output e.field
82 save outf=vds.str
83
84 # Solve Id versus reverse Vg curve
85
```

```
86
87  log outf=tftex05_1.log
88  solve vgate=0 vstep=-0.5 vfinal=-20. name=gate
89  save outf=tftex05_1.str
90  tonyplot  tftex05_1.str -set tftex05_1.set
91
92  # extract the peak reverse current
93  extract name="max_rev_id" y.val from curve(v."gate",i."drain") where
      x.val=-20
94
95  # Solve Id versus forward Vg curve
96
97  # restart from zero gate bias
98  load inf=vds.str master
99
100 log outf=tftex05_2.log
101 solve vgate=0 vstep=0.5 vfinal=20. name=gate
102
103 # extract subthreshold slope
104 extract name="subvtslope" 1.0/slope(maxs-
      lope(curve(abs(v."gate"),log10(abs(i."drain")))))
105
106 tonyplot -overlay tftex05_1.log tftex05_2.log -set tftex05_log.set
107
108 quit
```

15.1.6 tftex06.in : 3D TFT Simulation

Requires: DEVICE3D/TFT3D

This example demonstrates Ids/Vgs and Id/Vds 3D analysis of a 10um channel length, ultra-thin TFT transistor. The example shows the formation of 3D structure using ATLAS syntax

```
Id/Vgs solution with Vds=0.1V
Id/Vds solution with Vgs=1.5V
```

The formation of this 3D structure is performed using the ATLAS syntax. The syntax used is very similar to that used to create a normal 2D example. The definition of dimensions in the third direction is defined by the z indicator. Thus, `z.min` and `z.max` define extents in the z direction, just as x.min and x.max do in the x direction.

The TFT device is composed of a 0.05 micron layer of polysilicon on a 0.45 micron silicon dioxide substrate. The device has a 12 nm thick gate oxide and a gate length of 10 microns. The gate width is 80um.

After the device description the model statement is used to select a set of physical models for this simulation. In this case, these models are 'srh' recombination, bandgap narrowing, and the conmob and fldmob mobility models. The impact statement is used to specify the Selberherr model for impact ioisation in the 3D simulation. This is necessary to model "kink" effects that are often observed in TFT Id-Vd characteristics. The contact statement is used to assign the work function on the gate

material. It this case an aluminum gate is used which assigns a workfunction of 4.10 V to the gate contact.

The numerical methods used are also similar to the previous example: `METHOD gummel newton carr=2`. This means that if convergence is not reached in decoupled mode (gummel) the simulator will automatically switch to coupled mode (newton). In addition, a two carrier solution is performed by solving Poisson's equation for potential and the electron and hole continuity equations.

The drain voltage is set to 0.1V, and then the gate voltage is swept to 1.5V to measure the Ids/Vgs curve.

Then the gate voltage is set to 1.5V, and then the drain voltage is swept to 3V. At this point faster solutions are obtained using the coupled newton algorithm so the simulation switches to this method. The drain voltage is then ramped to 4V.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

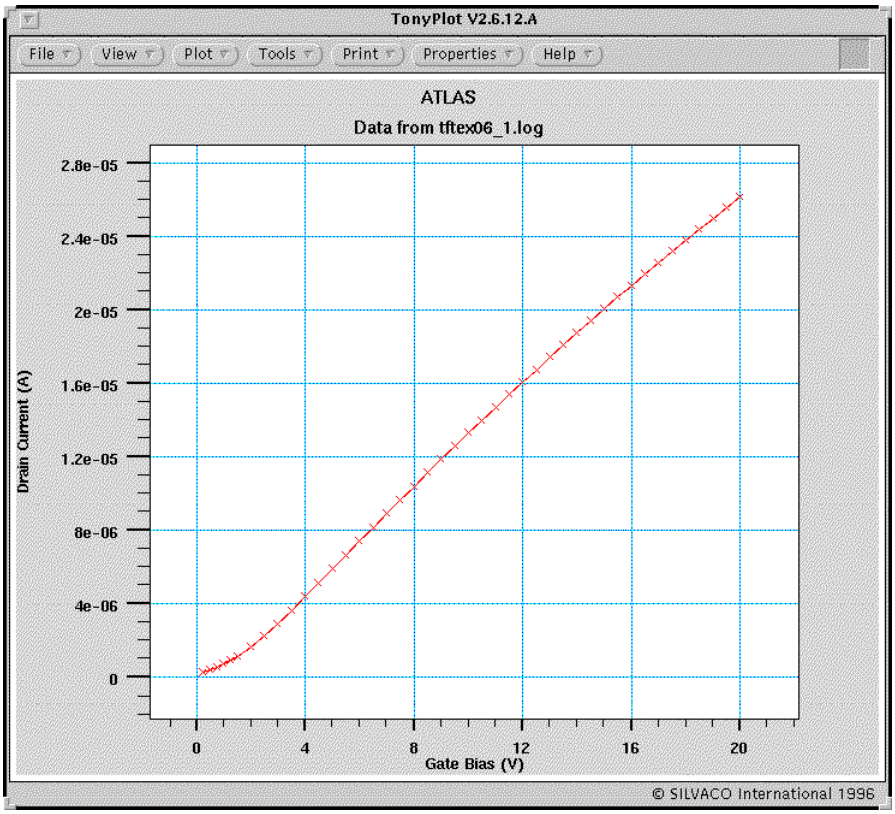


Figure 15.12: Drain current versus gate voltage of a 3D TFT transistor

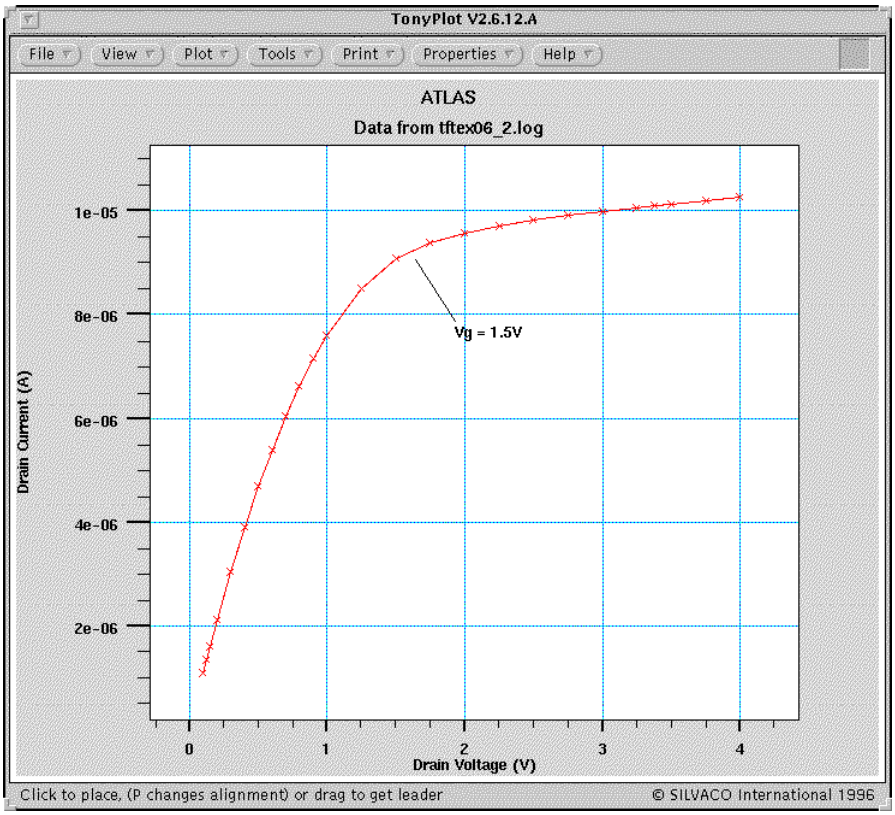


Figure 15.13: Drain current versus drain voltage of a 3D TFT transistor

Input Deck tft/tftex06.in :

```
1   go atlas
2   TITLE   a-si TFT simulation
3   # SILVACO International 1993, 1994
4
5   meshthree.d
6   #
7   x.ml=0spac=0.5
8   x.ml=2spac=0.5
9   x.ml=3.5spac=1.0
10  x.ml=5spac=0.25
11  x.ml=10spac=1.5
12  x.ml=15.spac=0.25
13  x.ml=16.5spac=1.0
14  x.ml=18spac=0.5
15  x.ml=20spac=0.5
16  #
17  #
18  y.ml=-0.1spac=0.05
19  y.ml=0spac=0.0075
```

```
y.ml=0.05spac=0.01
y.ml=0.1spac=0.0075
y.ml=0.5spac=0.25
y.ml=10.spac=5
#
#
z.ml=0spac=5
z.ml=15spac=5
#
#
#  *************** regions ***************
#             1=oxide  2=silicon 3-oxide
#
region       num=1  y.max=0.    oxide
region       num=2  y.min=0.    y.max=0.1   silicon
region       num=3  y.min=0.1   oxide
#
#  *************** electrodes ***************
#    1=gate  2=substrate  3=source  4=drain
#
elec  num=1  x.min=5    x.max=15  y.min=-0.1 y.max=-0.1 name=gate
elec  num=2  x.min=0.   x.max=2.    y.min=0.   y.max=0.   name=source
elec  num=3  x.min=18.0 x.max=20.0  y.min=0.   y.max=0.   name=drain
elec  num=4  substrate              name=substrate
#
#  *************** doping profiles  ***************
#
doping     reg=2  uniform conc=7.e14 n.type
doping     reg=2  gauss conc=3.e18 n.type x.right=5 char=0.3
doping     reg=2  gauss conc=3.e18 n.type x.left=15 char=0.3
#
#   Set parameters for amorphus silicon
#
material region=2 mun=20 mup=1.5 nc300=2.5e20 \
 nv300=2.5e20 eg300=1.9
#
defects nta=1.e21 ntd=1.e21 wta=0.033 wtd=0.049 \
  nga=1.5e15 ngd=1.5e15 ega=0.62 egd=0.78 wga=0.15 wgd=0.15 \
  sigtae=1.e-17 sigtah=1.e-15 sigtde=1.e-15 sigtdh=1.e-17  \
  siggae=2.e-16 siggah=2.e-15 siggde=2.e-15 siggdh=2.e-16
#
contactnum=1 alum
models srh bgn conmob fldmob temp=300
```

```
#
impact selber
#
method itlimit=30
solve init
#
save outf=tftex06_0.str
#
# Id-Vg calculations
#
#ramp the drain to 0.1V
#
method  bicgst  maxtrap=6  trap
solve vdrain=0.025
solve vdrain=0.05
solve vdrain=0.1
#
#ramp the gate and store the results
#
log outf=tftex06_1.log
#
solve vstep=0.25 vfinal=1.5 name=gate
#
save outf=tftex06_1.str
#
solve vgate=2.0 vstep=0.5 vfinal=20 name=gate
#
# Id-Vd calculations
#
#ramp the gate to 1.5V
#
log off
#
method  bicgst carr=2  maxtrap=6  trap
#
#ramp the drain and store the results
#
load infile=tftex06_1.str master
#
log outf=tftex06_2.log master
solve prev
solve vdrain=0.1 vstep=0.1 vfinal=1 name=drain
solve vdrain=1.25 vstep=0.25 vfinal=3 name=drain
```

```
106 solve vdrain=3.25 vstep=0.25 name=drain vfinal=4
107 #
108 tonyplot tftex06_1.log -set tftex06_1.set
109 tonyplot tftex06_2.log -set tftex06_2.set
110 quit
```

15.1.7 tftex07.in : Comparing a Discrete and Continuous Distribution of Traps

Requires: SPISCES/TFT

It is possible to specify an energy dependent distribution of traps in two ways using SPISCES/TFT. The original method assumed a discrete set of traps across the bandgap but now it is possible to use a mathematically continuous energy distribution of states.

The syntax used in this example is identical to that in `tftex03.in` except that now two simulations are performed each with the density of states assumed to contain 24 energy levels across the bandgap. This is speciied using the `Defect` parameters `numa` and `numd`. The first simulation is otherwise identical to that described in `tftex03.in`. The second simulation uses the keyword continuous on the `Defects` statement to define that the distribution is continuous. The simulation then compares the Id-Vg characteristics for the two simulations.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

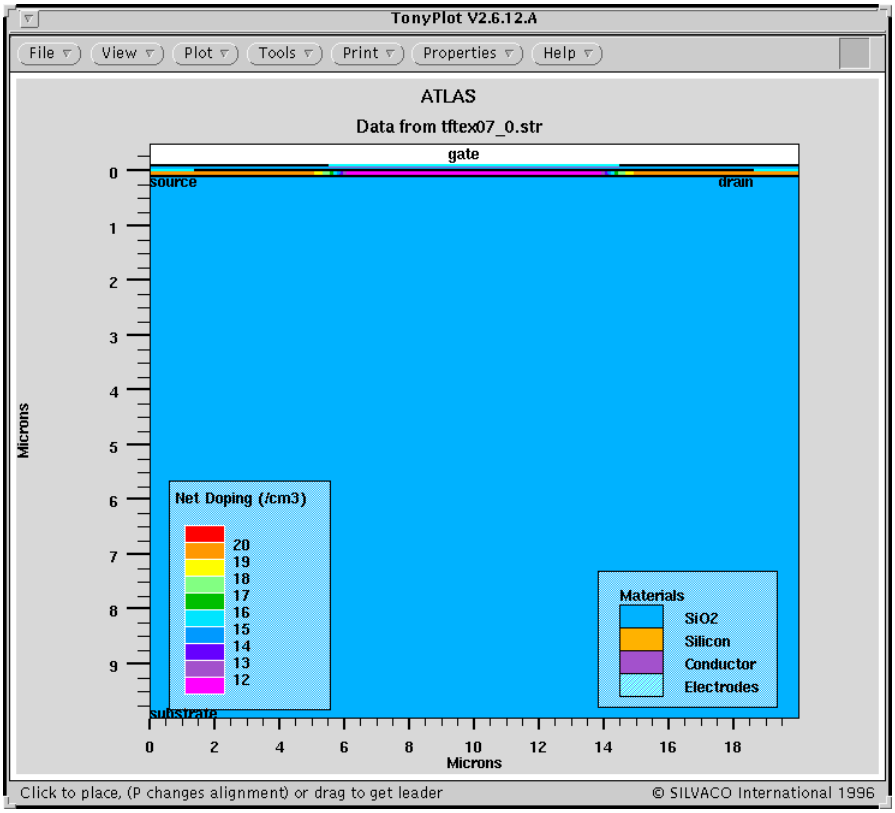


Figure 15.14: Geometry and Doping of a TFT transistor

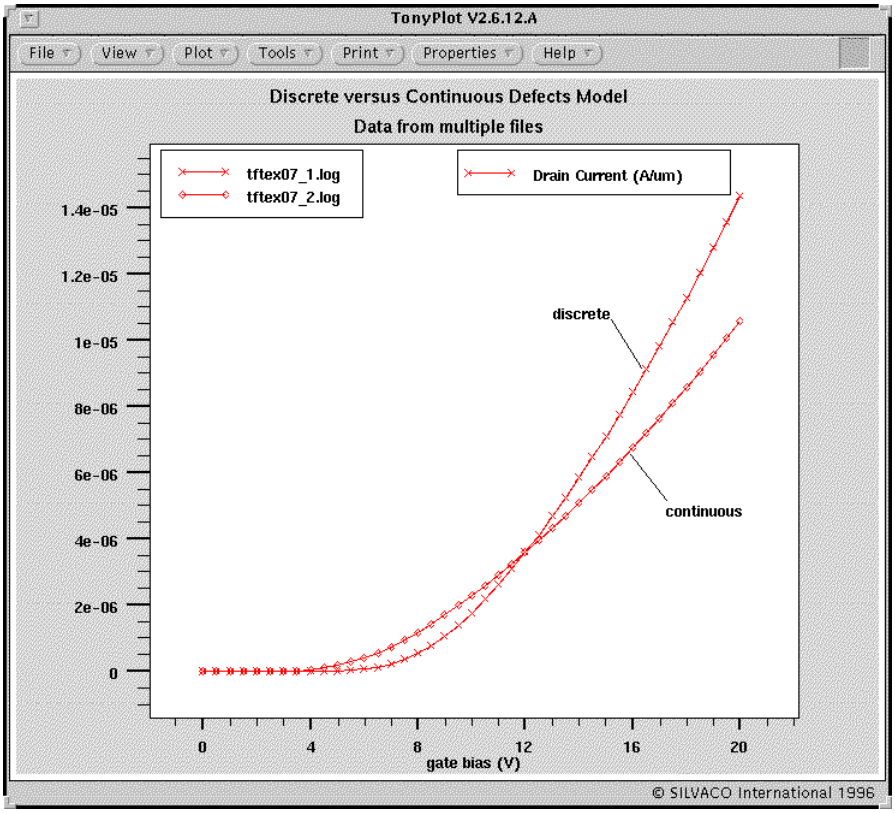


Figure 15.15: Linear region of the Id - Vg characteristic of a TFT transistor showing the difference between the discrete and continuous TFT DOS model

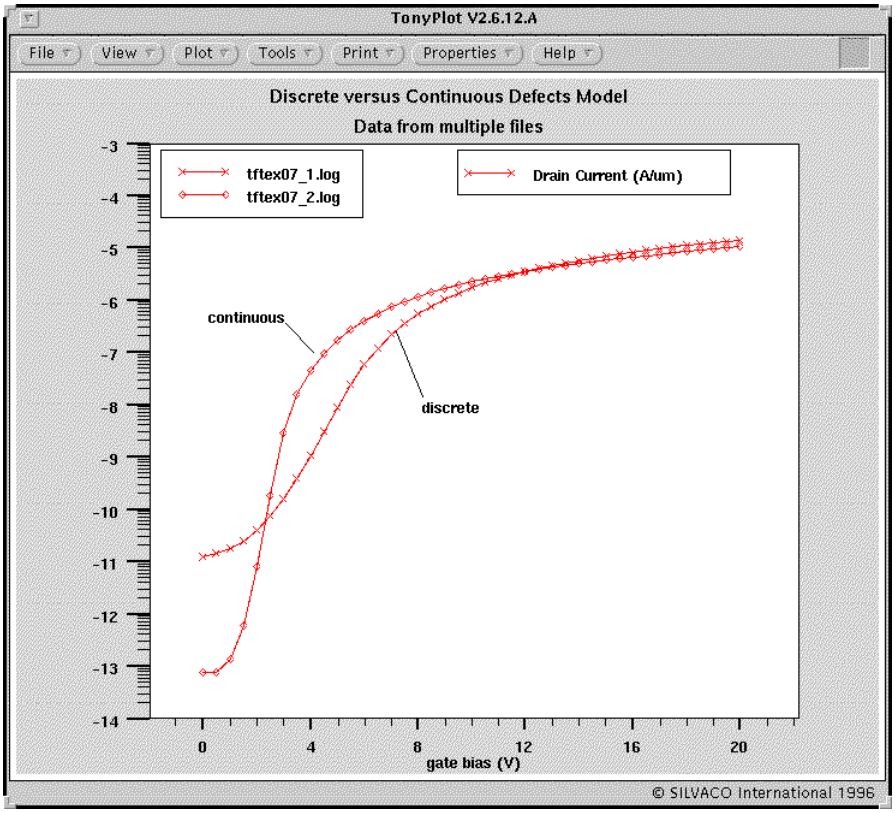


Figure 15.16: Subthreshold region of the Id - Vg characteristic of a TFT transistor showing the difference between the discrete and continuous TFT DOS model

Input Deck tft/tftex07.in :

```
go atlas
TITLE polysilicon TFT simulation
# SILVACO International 1993, 1994

mesh   nx=30  ny=30
#
x.m          n=1      l=0          r=1
x.m          n=30     l=20.        r=1
#
#
y.m          n=1      l=-0.1    r=1
y.m          n=5      l=0       r=1
y.m          n=19     l=0.1     r=1
y.m          n=30     l=10.     r=1.6
#
#
#  ***************** regions *******************************
#             1=oxide  2=silicon 3-oxide
```

```
#
region       num=1  y.max=0.    oxide
region       num=2  y.min=0. y.max=0.1   silicon
region       num=3  y.min=0.1   oxide
#
#  ***************** electrodes ****************************
#    1=gate  2=substrate  3=source  4=drain
#
elec  num=1  x.min=5   x.max=15  y.min=-0.1 y.max=-0.1 name=gate
elec  num=2  substrate            name=substrate
elec  num=3  x.min=0.  x.max=2.  y.min=0.   y.max=0.   name=source
elec  num=4  x.min=18. x.max=20. y.min=0.   y.max=0.   name=drain
#
#  ***************** doping profiles ***********************
#
doping     reg=2  uniform conc=1.e11 n.type
doping     reg=2  gauss conc=1.e20 n.type x.right=5 char=0.3
doping     reg=2  gauss conc=1.e20 n.type x.left=15 char=0.3

#
#
#     Set parameters for polysilicon
#
material region=2 mun=300 mup=30
#
# Use the discrete defect model with 24 levels
defects  numa=24 numd=24 \
  nta=1.12e21 ntd=4.e20 wta=0.025 wtd=0.05 \
  nga=5.e17 ngd=1.5e18 ega=0.4 egd=0.4 wga=0.1 wgd=0.1 \
  sigtae=1.e-16 sigtah=1.e-14 sigtde=1.e-14 sigtdh=1.e-16  \
  siggae=1.e-16 siggah=1.e-14 siggde=1.e-14 siggdh=1.e-16
#
contactnum=1 n.polysilicon
models  temp=300
#
method newton

solve init
save outf=tftex07_0.str
tonyplot tftex07_0.str -set tftex07_0.set


#
```

```
# Electrode #1 - Gate, #2 - Substrate, #3 - Source, #4 - Drain
#

method newton
#
solve  vdrain=0.1
solve  vdrain=0.2
solve  vdrain=0.5
solve  vdrain=1 vstep=1. vfinal=5. name=drain
#
# Solve id versus vg curve
#

#
log outf=tftex07_1.log
solve vstep=0.5 vfinal=20 name=gate vgate=0. vdrain=5

go atlas
TITLE polysilicon TFT simulation
# SILVACO International 1993, 1994

mesh   nx=30  ny=30
#
x.m          n=1      l=0         r=1
x.m          n=30     l=20.       r=1
#
#
y.m          n=1      l=-0.1    r=1
y.m          n=5      l=0       r=1
y.m          n=19     l=0.1     r=1
y.m          n=30     l=10.     r=1.6
#
#
#  ***************** regions *******************************
#             1=oxide  2=silicon 3-oxide
#
region       num=1  y.max=0.    oxide
region       num=2  y.min=0. y.max=0.1   silicon
region       num=3  y.min=0.1   oxide
#
#  ***************** electrodes ****************************
#    1=gate  2=substrate  3=source  4=drain
#
```

```
105 elec  num=1  x.min=5   x.max=15  y.min=-0.1 y.max=-0.1 name=gate
106 elec  num=2  substrate            name=substrate
107 elec  num=3  x.min=0.  x.max=2.  y.min=0.   y.max=0.   name=source
108 elec  num=4  x.min=18. x.max=20. y.min=0.   y.max=0.   name=drain
109 #
110 #  ***************** doping profiles ***********************
111 #
112 doping     reg=2  uniform conc=1.e11 n.type
113 doping     reg=2  gauss conc=1.e20 n.type x.right=5 char=0.3
114 doping     reg=2  gauss conc=1.e20 n.type x.left=15 char=0.3
115
116 #
117 #
118 #     Set parameters for polysilicon
119 #
120   material region=2 mun=300 mup=30
121 #
122 # Use the continuous defect model
123   defects continuous numa=24 numd=24 \
124   nta=1.12e21 ntd=4.e20 wta=0.025 wtd=0.05 \
125   nga=5.e17 ngd=1.5e18 ega=0.4 egd=0.4 wga=0.1 wgd=0.1 \
126   sigtae=1.e-16 sigtah=1.e-14 sigtde=1.e-14 sigtdh=1.e-16  \
127   siggae=1.e-16 siggah=1.e-14 siggde=1.e-14 siggdh=1.e-16
128 #
129 contactnum=1 n.polysilicon
130 models  temp=300
131 #
132 method newton
133
134 solve init
135
136
137 #
138 # Electrode #1 - Gate, #2 - Substrate, #3 - Source, #4 - Drain
139 #
140
141 method newton
142 #
143 solve  vdrain=0.1
144 solve  vdrain=0.2
145 solve  vdrain=0.5
146 solve  vdrain=1 vstep=1. vfinal=5. name=drain
147 #
```

```
148 # Solve id versus vg curve
149 #
150
151 #
152 log outf=tftex07_2.log
153 solve vstep=0.5 vfinal=20 name=gate vgate=0. vdrain=5
154
155 tonyplot -overlay tftex07_1.log tftex07_2.log -set tftex07_1.set
156 tonyplot -overlay tftex07_1.log tftex07_2.log -set tftex07_2.set
157 #
158 quit
159
160
```

15.1.8 tftex08.in : TFT Simulation Using the C-interpreter

Requires: SPISCES/TFT/C-INTERPERTER

The `Defects` statement only allows the user to specify a fixed density of states versus energy distribution. This would normally comprise one Gaussian and/or one exponential energy distribution of traps across the bandgap. This may not necessarily accurately reflect the users material.

However, ATLAS allows the user to use the `C-interpreter` to specify the density of states versus energy distribution. On the `Defects` statement, the options f.tftacc and f.tftdon specify files which should contain the distributions written as standard `C` routines.

In this example the acceptor and donor densities are defined by two exponential equations each which begin from the conduction and the valence bands. The parameters that would normally specify the density of states should be set to `nta=ntd=nga=ngd=0`, otherwise they would be added to the `C-interpreter` values.

The capture cross sections are then taken from the parameters in the `Defects` statement. The acceptor capture cross section is defined by the Gaussian acceptor value and the donor capture cross section is defined by the Gaussian donor value.

The remainder of this TFT input deck is the same as that described in `tftex03.in`.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

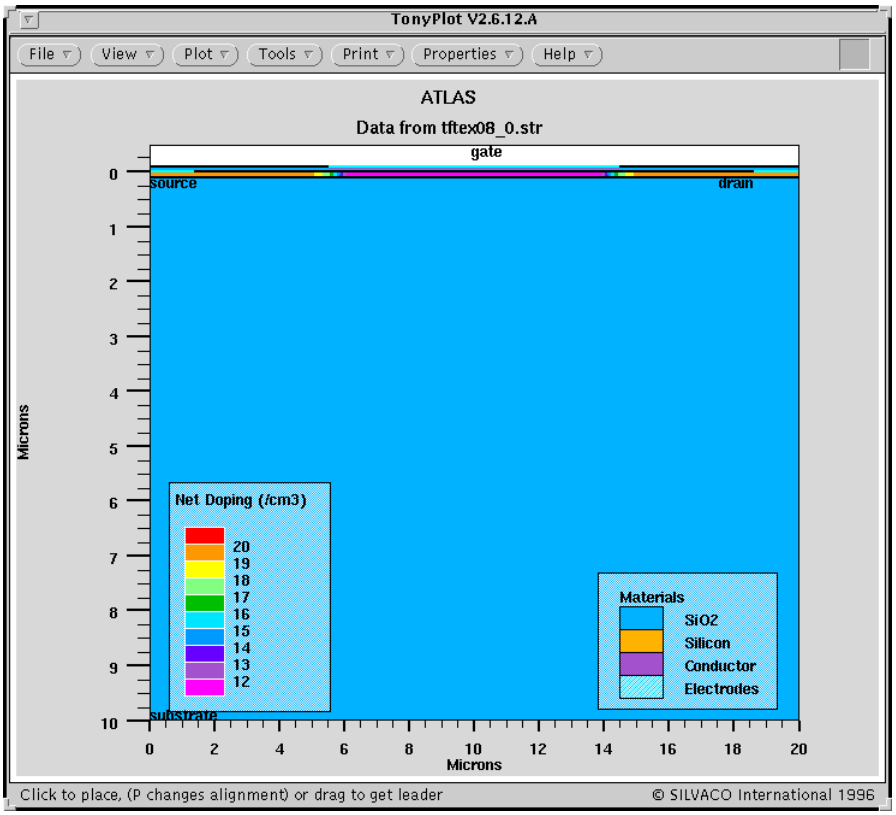


Figure 15.17: Geometry and Doping of a TFT transistor

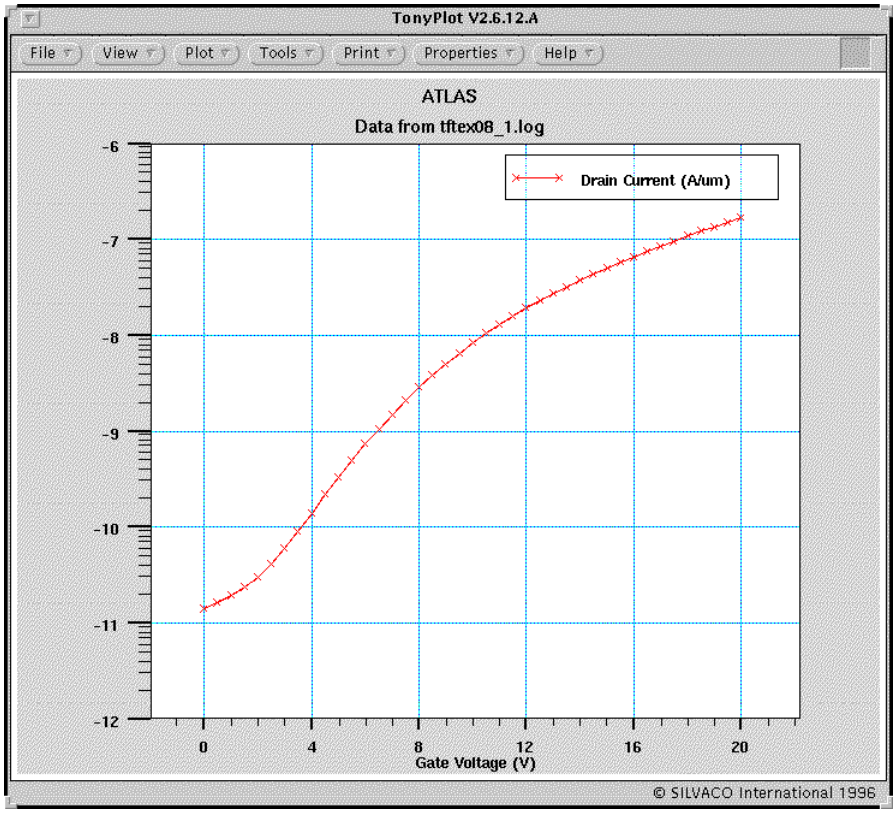


Figure 15.18: Id-Vg characteristic with the DOS derived from a C-Interpreter function

Input Deck tft/tftex08.in :

```
1   go atlas
2   TITLE polysilicon TFT simulation
3   # SILVACO International 1993, 1994
4
5   mesh   nx=30  ny=30
6   #
7   x.m          n=1      l=0          r=1
8   x.m          n=30     l=20.        r=1
9   #
10  #
11  y.m          n=1      l=-0.1     r=1
12  y.m          n=5      l=0        r=1
13  y.m          n=19     l=0.1      r=1
14  y.m          n=30     l=10.      r=1.6
15  #
16  #
17  #  *************** regions ******************************
18  #             1=oxide  2=silicon 3-oxide
19  #
```

```
region       num=1  y.max=0.    oxide
region       num=2  y.min=0. y.max=0.1   silicon
region       num=3  y.min=0.1   oxide
#
#  *************** electrodes ***************************
#    1=gate  2=substrate  3=source  4=drain
#
elec  num=1  x.min=5   x.max=15  y.min=-0.1 y.max=-0.1 name=gate
elec  num=2  substrate            name=substrate
elec  num=3  x.min=0.  x.max=2.  y.min=0.   y.max=0.   name=source
elec  num=4  x.min=18. x.max=20. y.min=0.   y.max=0.   name=drain
#
#  *************** doping profiles ***********************
#
doping     reg=2  uniform conc=1.e11 n.type
doping     reg=2  gauss conc=1.e20 n.type x.right=5 char=0.3
doping     reg=2  gauss conc=1.e20 n.type x.left=15 char=0.3

#
#
#     Set parameters for polysilicon
#
material region=2 mun=300 mup=30
#
# Use the discrete defect model
defects  f.tftacc=tftex08.lib f.tftdon=tftex08.lib  \
  nta=0 ntd=0 wta=0.025 wtd=0.05 \
  nga=0 ngd=0 ega=0.4 egd=0.4 wga=0.1 wgd=0.1 \
  sigtae=1.e-16 sigtah=1.e-14 sigtde=1.e-14 sigtdh=1.e-16  \
  siggae=1.e-16 siggah=1.e-14 siggde=1.e-14 siggdh=1.e-16  \
  afile=acceptor.dat dfile=donor.dat
#
contactnum=1 n.polysilicon
models  temp=300
#
method newton

solve init
save outf=tftex08_0.str
tonyplot tftex08_0.str -set tftex08_0.set


#
```

```
63  # Electrode #1 - Gate, #2 - Substrate, #3 - Source, #4 - Drain
64  #
65
66  method newton
67  #
68  solve  vdrain=0.1
69  solve  vdrain=0.2
70  solve  vdrain=0.5
71  solve  vdrain=1 vstep=1. vfinal=5. name=drain
72  #
73  # Solve id versus vg curve
74  #
75
76  #
77  log outf=tftex08_1.log
78  solve vstep=0.5 vfinal=20 name=gate vgate=0. vdrain=5
79
80  tonyplot tftex08_1.log -set tftex08_1.set
81  #
82  quit
83
84
```

15.1.9 tftex09.in : TFT Simulation of Grains and Grain Boundaries

Requires: SPISCES/TFT

This example demonstrates how ATLAS and SPISCES may be used to model the influence of grain boundaries on the device simulation results. The input deck has been created in such a way as to allow easy implementation into a VWF automation tools environment.

The input deck initially uses the internal simulator within deckbuild to define a number of variables using the deckbuild set statement syntax. There are six variables the the user is required to specify which define the physical geometry of the TFT. The other variables are automatically defined and are used to create the mesh, regions and electrodes.

The grain and grain boundary are created by defining two regions in the polysilicon. Each region then has it's own material and defect statements that define the properties within the grain and the grain boundary.

The remainder of the syntax is as normal and is described in the previous TFT examples.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

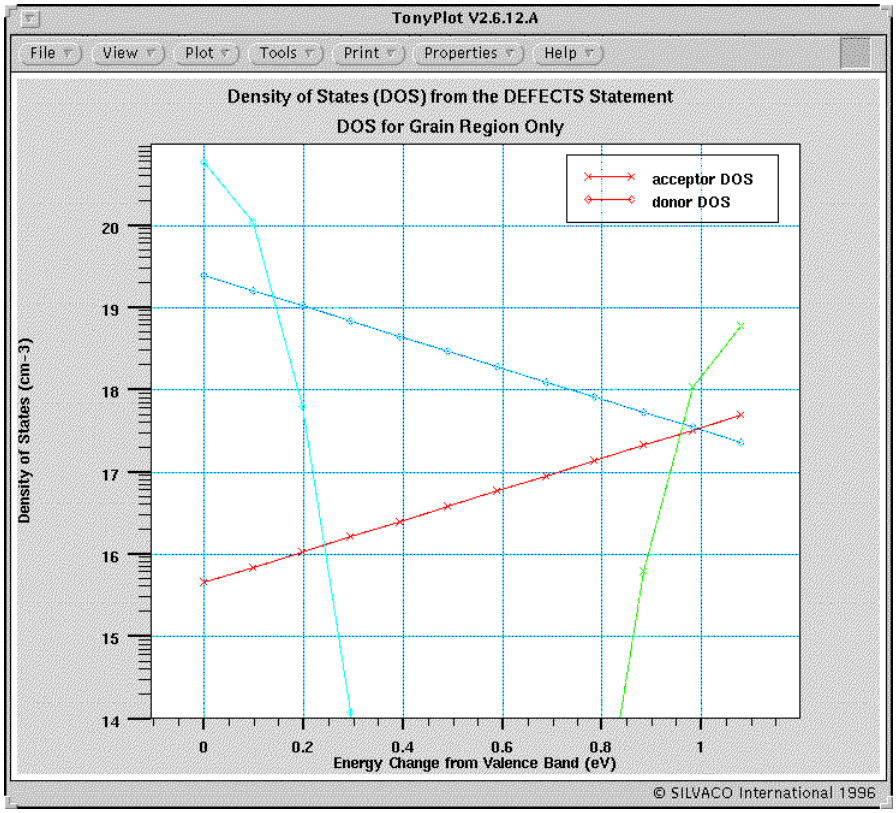


Figure 15.19: Density-of-States profile applied within the polysilicon grain of the TFT

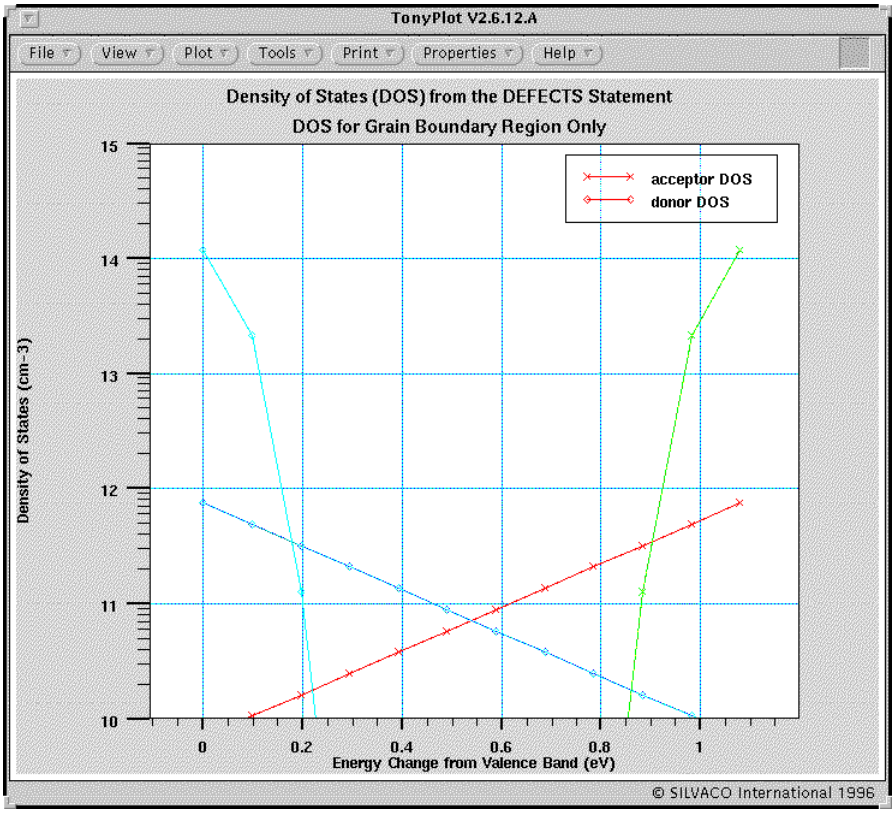


Figure 15.20: Density-of-States profile applied within the polysilicon grain boundary of the TFT

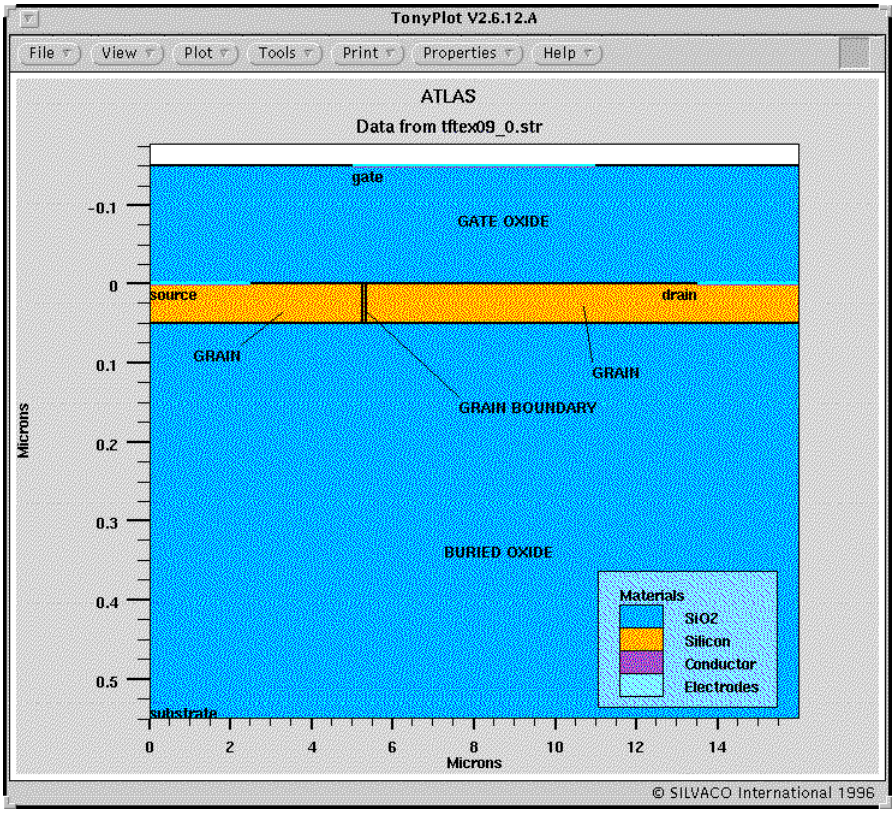


Figure 15.21: Geometry of TFT showing the location of the grain and one grain boundary

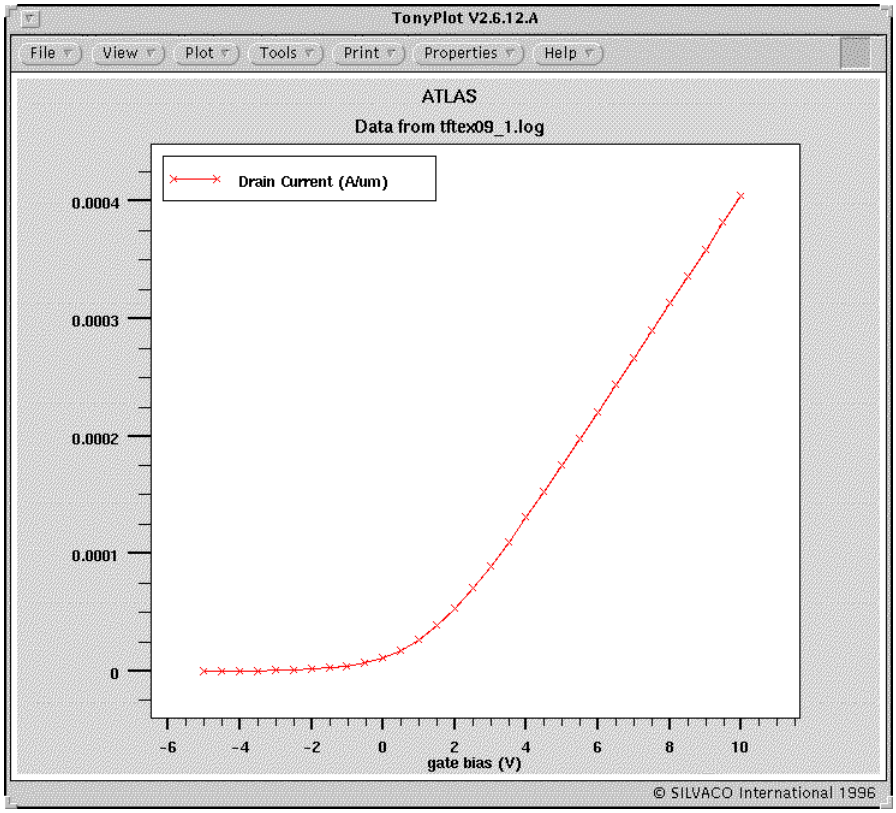


Figure 15.22: Threshold voltage characteristic of the TFT with both grains and grain boundary

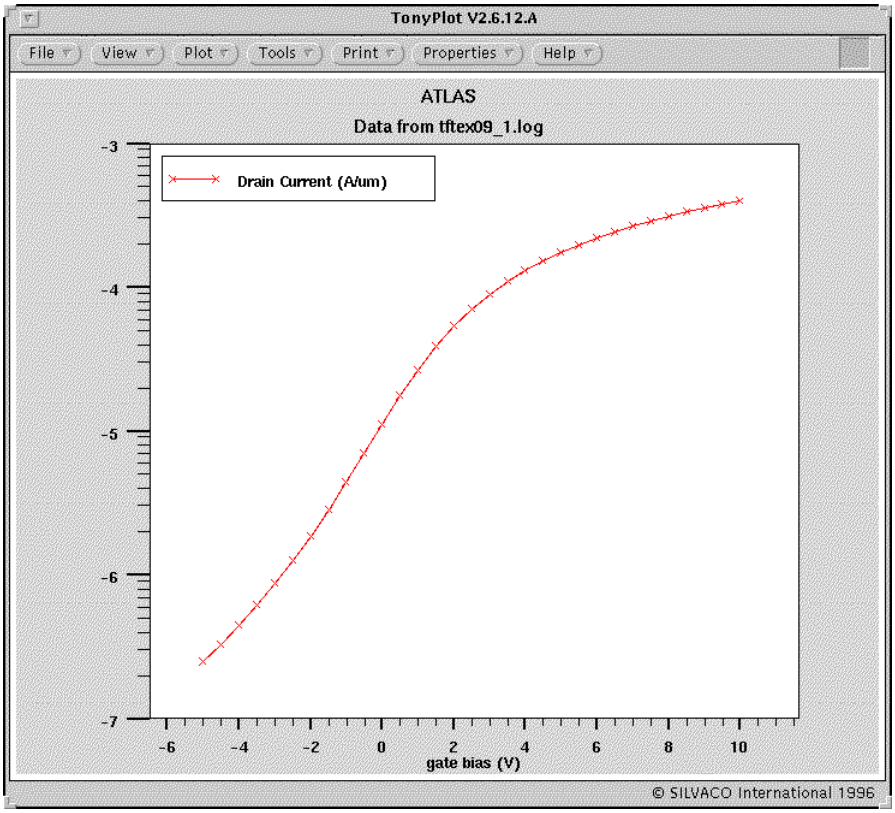


Figure 15.23: Subthreshold of Id-Vg characteristic of the TFT with both grains and grain boundary

Input Deck tft/tftex09.in :

```
1   go internal
2
3   #     Rules for Position of Grain Boundary
4   #     nlength < GrainStart+GrainLength < nlength+Lg
5   #
6
7   #
8   # Set up 7 Device Specifications
9   #
10  set nlength=5
11  set GateLength=6
12  set GrainLength=0.1
13  set GrainDeltaX=0.25
14  set SiliconThickness=0.05
15  set GateOxideThickness=0.15
16  set BuriedOxideThickness=0.5
17
18  # The following variables are used to automatically
19  # create the mesh, regions, electrodes and  doping.
```

```
#
set HalfGateOxideThickness=$"GateOxideThickness"/2
set HalfGateLength=$"GateLength"/2

set GateStart=$"nlength"
set GateEnd=$"GateStart" + $"GateLength"
set GateSpace=$"GateLength" / 12
set DrainGateSpace=$"GateSpace" / 2

set GrainStart=$"GateStart"+$"GrainDeltaX"
set GrainEnd=$"GrainStart"+$"GrainLength"
set GrainSpace=$"GrainLength"/5

set HalfSiliconThickness=$"SiliconThickness"/2
set SiliconThicknessSpace=$"SiliconThickness"/100
set HalfSiliconThicknessSpace=$"SiliconThickness"/10

set DeviceStart=0
set DeviceStartSpace=$"nlength"/10
set DeviceEnd=$"DeviceStart" + $"GateEnd" +$"nlength"
set DeviceBottom=$"SiliconThickness"+$"BuriedOxideThickness"
set DeviceBottomSpace=$"BuriedOxideThickness"/2

set SourceEnd=$"nlength"/2
set DrainStart=$"DeviceEnd"-$"nlength"/2

go atlas

mesh width=100

x.m l=0 s=$"DeviceStartSpace"
x.m l=$"SourceEnd" s=$"DeviceStartSpace"
x.m l=$"GateStart" s=$"GateSpace"
x.m l=$"GrainStart" spac=$"GrainSpace"
x.m l=$"GrainEnd" s=$"GrainSpace"
x.m l=$"GateEnd" s=$"GateSpace"
x.m l=$"DrainStart" s=$"DeviceStartSpace"
x.m l=$"DeviceEnd" s=$"DeviceStartSpace"
#
y.m l=-$"GateOxideThickness" s=$"HalfGateOxideThickness"
y.m l=0 s=$"SiliconThicknessSpace"
y.m l=$"HalfSiliconThickness" s=$"HalfSiliconThicknessSpace"
y.m l=$"SiliconThickness" s=$"HalfSiliconThicknessSpace"
```

```
y.m l=$"DeviceBottom" s=$"DeviceBottomSpace"
#
region num=1 oxide name=box
region num=2 silicon y.min=0 y.max=$"SiliconThickness"
region num=3 silicon x.min=$"GrainStart" x.max=$"GrainEnd" y.min=0
  y.max=$"SiliconThickness"
#
electrodes num=1 name=source x.max=$"SourceEnd" y.min=0 y.max=0
electrodes num=2 name=gate x.min=$"GateStart" x.max=$"GateEnd" y.max=-
  $"GateOxideThickness"
electrodes num=3 name=drain x.min=$"DrainStart" y.min=0 y.max=0
electrodes num=4 name=substrate bottom
#
doping  uniform conc=1.e15 n.type
doping  gauss conc=3.e18 n.type x.right=$"GateStart" char=0.3
doping  gauss conc=3.e18 n.type x.left=$"GateEnd" char=0.3
#
#   Set parameters for polysilicon
#
material region=2 mun=300 mup=30 taun0=1e-8 taup0=1e-8
material region=3 mun=30 mup=3 taun0=1e-10 taup0=1e-10
#
defects region=2 continuous afile=tftex09_acc1.dat dfile=tftex09_don1.dat
  \
  nga=6e18 ngd=6e20  ega=1.08 egd=0.0 wga=0.075 wgd=0.075 \
  nta=5e17 ntd=2.5e19 wta=0.23 wtd=0.23 \
  sigtae=1.e-16 sigtah=1.e-14 sigtde=1.e-14 sigtdh=1.e-16  \
  siggae=1.e-16 siggah=1.e-14 siggde=1.e-14 siggdh=1.e-16
#
defects region=3 continuous afile=tftex09_acc2.dat dfile=tftex09_don2.dat
  \
  nga=1.2e14 ngd=1.2e14  ega=1.08 egd=0.0 wga=0.075 wgd=0.075 \
  nta=7.5e11 ntd=7.5e11 wta=0.23 wtd=0.23 \
  sigtae=1.e-16 sigtah=1.e-14 sigtde=1.e-14 sigtdh=1.e-16  \
  siggae=1.e-16 siggah=1.e-14 siggde=1.e-14 siggdh=1.e-16
#
contactnum=2 alum
models srh temp=300 print
#
method itlimit=30

tonyplot -overlay tftex09_acc1.dat tftex09_don1.dat -set tftex09_0.set
tonyplot -overlay tftex09_acc2.dat tftex09_don2.dat -set tftex09_1.set

```

```
103
104 solve init
105 save outf=tftex09_0.str
106 tonyplot tftex09_0.str -set tftex09_2.set
107 #
108 solve prev
109 #
110 # Electrode #1 - Source, #2 - Gate, #3 - Drain, #4 - Substrate
111 #
112 method  gummel newton gum.init=6
113 solve  vgate=-5.0
114
115 method  newton
116 solve  vdrain=0.1
117 solve  vdrain=0.2
118 solve  vdrain=0.5
119 #
120 # Solve id versus vg curve
121 #
122 log outf=tftex09_1.log
123 solve prev
124 solve prev vstep=0.5 vfinal=10. name=gate
125 #
126 tonyplot tftex09_1.log -set tftex09_3.set
127 tonyplot tftex09_1.log -set tftex09_4.set
128 quit
```

This page is intentionally left blank

16.1. OPTOELECTRONICS: Optoelectronics Application Examples

16.1.1 optoex01.in: Photodetector – Overview and Structure Definition

Requires: S-PISCES

The following set of examples demonstrate how to extract device characteristics for general photodetectors. These characteristics are of interest to those involved in the design of PN and PIN photodiodes, avalanche photodiodes, photoconductors, phototransistors, optical FETs and MSMs. These examples focus principally on optoelectronic characteristics as opposed to purely electronic behavior of photodetectors (e.g. dark current, capacitance).

The following examples will address a `PIN` photodetector. The extraction techniques used on the PIN device will equally apply to other photodetector devices. The device for these examples is described in the input file `optoex01.in`. This input file should be run before any of the other examples in this section as it provides the base device structure used in the other examples.

The first section of the input file specifies the mesh. In this case, the mesh is ten microns by ten microns. The spacings were chosen to help resolve the `P+ N` and `N N+` junctions. The `space.mult` parameter is used to make the mesh coarse for fast simulation. For more accuracy, the value should be set to unity.

The second section of the input file specifies the structure of the device. The device is a square of silicon with an anode extending across the front surface (`y=0.0`) and a cathode across the bottom. It is uniformly doped N-type at 1.0e14, with heavily doped P+ and N+ regions at the front and back surfaces respectively.

The third section of the input file specifies which material models will be used in the simulation. Here, SRH and Auger recombination mechanisms have been specified, since they have important effects on the quantum efficiency of photodiodes. Concentration and field dependent mobility models are also specified.

The last section of the input file is used to obtain an initial solution. The initial solution is written to a solution file that is used in all the subsequent photodetector analysis examples. The output file can be viewed using TONYPLOT. This shows the structure of the device and the device mesh.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

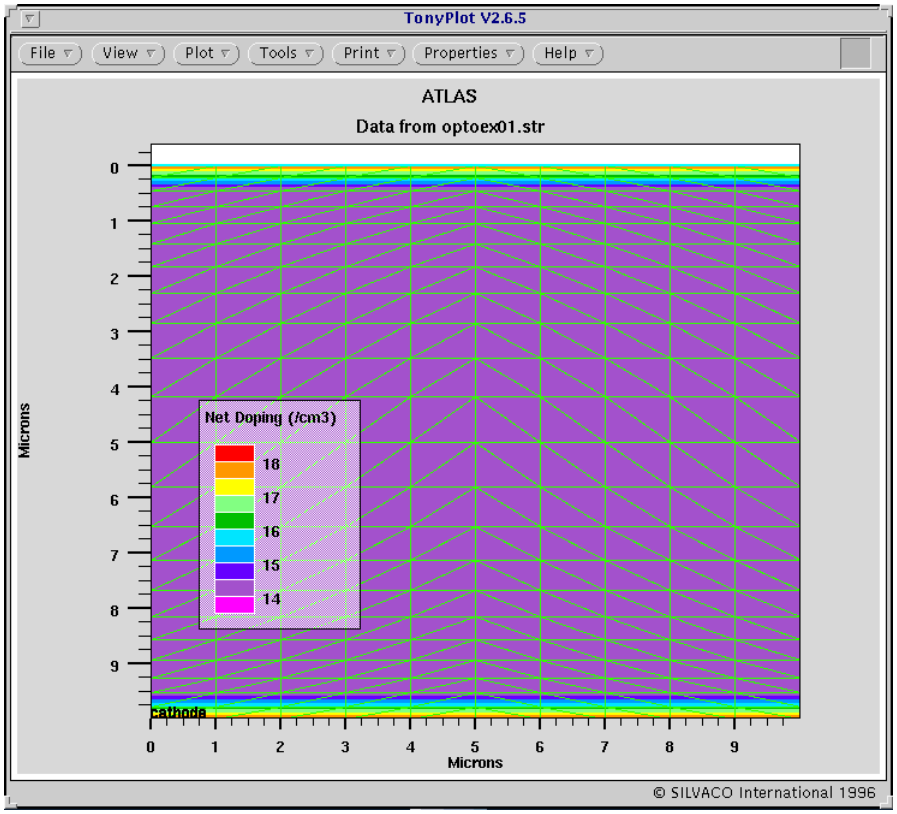


Figure 16.1: Doping and Mesh of pin photodetector defined using ATLAS syntax

Input File optoelectronics/optoex01.in:

```
1   go atlas
2   Title PIN photodiode simulation example
3   #
4   # PIN device description and initial solution
5   # SILVACO International 1992, 1993, 1994, 1995
6   #
7   # SECTION 1: Mesh Specification
8   #
9   mesh space.mult=4.0
10  #
11  x.mesh loc=0.0  spacing=0.25
12  x.mesh loc=10.0 spacing=0.25
13  #
14  y.mesh loc=0.0   spacing=0.05
15  y.mesh loc=5.0   spacing=0.2
16  y.mesh loc=10.0  spacing=0.05
17  #
18  # SECTION 2: Structure Specification
19  #
```

```
region num=1 material=Silicon
#
elec num=1 name=anode x.min=0.0 x.max=10.0 y.max=0.0
elec num=2 name=cathode bottom
#
doping uniform conc=1e14 n.type
doping gaus peak=0.0 char=0.1 conc=1e18 p.type dir=y
doping gaus peak=10.0 char=0.1 conc=1e18 n.type dir=y
#
# SECTION 3: Material Model Specification
#
material taup0=2.e-6 taun0=2.e-6
models srh auger conmob fldmob
#
# SECTION 4: Initial Solution
#


solve    init outf=optoex01.str master

tonyplot optoex01.str -set optoex01.set
quit
```

16.1.2 optoex02.in: Photodetector: DC Characterization

Requires: S-PISCES/LUMINOUS

The first example involves characterization of steady state characteristics of the device. This example demonstrates extraction of DC currents as a function of optical source power. This example also demonstrates the use of multi-spectral sources. The source spectrum is described in the file "optoex02.spec".

The first section of the input file specifies that the mesh and device structure is taken from the previous example. The second section of the file sets the same material models that were specified in the previous example.

The third section of the input file specifies the optic source. In this example, the source originates one micron above the device and is directed normal to the front surface of the device. The source spectrum is described in the file `optoex02.spec`. This spectrum is sampled at five discrete samples between the wavelengths of 0.5 and 0.8 microns.

The fourth section of the file sets up the initial operating bias for the simulation. In this case an operating bias of 2.0 volts is chosen.

The last section of the input file specifies that the optical source intensity will be ramped from 0 to 1 W/cm^2. During this ramp, steady state currents are extracted and saved in the log file.

The results are displayed by TONYPLOT. The first figure shows several curves. The first curve, source photo current, is the equivalent current that would be observed if all the light from the source were detected. The second curve, available photo current, is the equivalent current that would be observed if all the light absorbed were converted to terminal current. The difference between the source and available photo currents is due to part of the light passing all the way through the device.

The third curve is the cathode current. By taking the ration of the cathode current with the source photo current, the quantum efficiency as a function of source intensity is plotted in the second figure. Here the device quantum efficiency is about 91% and is relatively independent of source intensity.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

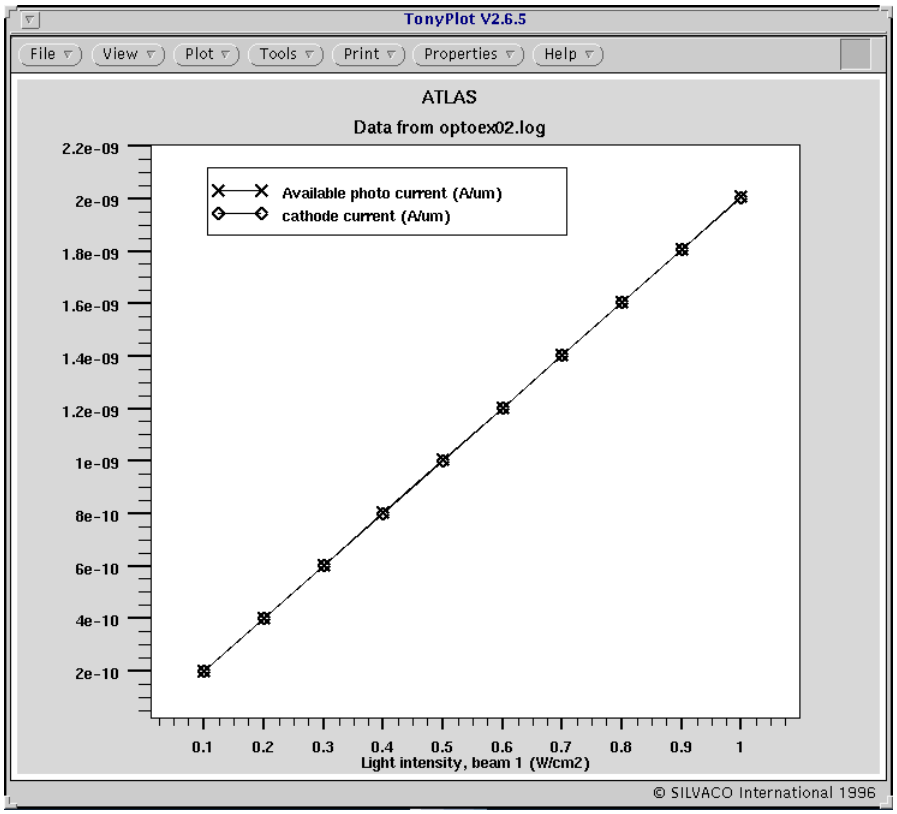


Figure 16.2: DC simulation of current output as a function of light intensity.

Input File optoelectronics/optoex02.in:

```
1   go atlas
2   Title PIN diode DC solutions
3   #
4   # PIN DC solution
5   # SILVACO International 1992, 1993, 1994, 1995
6
7   # PIN device description and initial solution
8   #
9   # SECTION 1: Mesh Specification
10  #
11  mesh space.mult=4.0
12  #
13  x.mesh loc=0.0  spacing=0.25
```

```
x.mesh loc=10.0 spacing=0.25
#
y.mesh loc=0.0   spacing=0.05
y.mesh loc=5.0   spacing=0.2
y.mesh loc=10.0  spacing=0.05
#
# SECTION 2: Structure Specification
#
region num=1 material=Silicon
#
elec num=1 name=anode x.min=0.0 x.max=10.0 y.max=0.0
elec num=2 name=cathode bottom
#
doping uniform conc=1e14 n.type
doping gaus peak=0.0 char=0.1 conc=1e18 p.type dir=y
doping gaus peak=10.0 char=0.1 conc=1e18 n.type dir=y
#
# SECTION 3: Material Model Specification
#
material taup0=2.e-6 taun0=2.e-6
models srh auger conmob fldmob
#
# SECTION 4: Initial Solution
#



solve    init outf=optoex02_0.str master

tonyplot optoex02_0.str -set optoex02_0.set

#
# SECTION 5: Optical source definition

# define a multi-spectral beam normal to top (y=0.0) surface
# this beam is as follows:
# beam #1, originating at (2.5,-1.0), propogating at an angle
# of 90 degrees, with a multi-spectral spectrum described in
# the file "optoex02.spec", sampled from 0.5 microns to 0.88 microns
# at 5 sample wavelengths.
#
beam     num=1 x.origin=5.0 y.origin=-1.0 angle=90.0 pow-
  er.file=optoex02.spec wavel.start=0.5 wavel.end=0.8 wavel.num=5

```

```
#
method   newton trap
solve init
solve    vcathode=0.1
solve    vcathode=0.5
solve    vcathode=1.0
solve    vcathode=2.0
#
# SECTION 6: Current vs. intensity
# step light by steps of 1 for 10 steps
#
log     outf=optoex02.log master
solve b1=0.1 lit.step=0.1 nstep=9 outf=optoex02_1.str master

tonyplot optoex02.log -set optoex02_1.set

quit
```

16.1.3 optoex03.in: Photodetector: Transient Response

Requires: S-PISCES/LUMINOUS

This example deals with time domain response of the device to time dependent optical sources. This example will extract the transient response of the device to an optical source being “turned off”.

The first and second sections of the input file are the same as in the DC example described previously. They specify the material models used and they specify that the device structure is read in from an external file and. The third section of the file specifies a normally incident monochromatic source. The wavelength of the source is 623 nm (equivalent to a HeNe laser wavelength).

The fourth section of the input file sets the initial biasing conditions of the device. Here again the device is biased to 2.0 volts.

The last section of the input file specifies the transient. In this case the DC solution at a source intensity of 5, W/cm^2 is obtained. Then, the source intensity is ramped linearly to 0 over a time of 1ns. Terminal data is then collected over the next 9ns. The terminal data is saved to the log file.

The results of the example are viewed in TONYPLOT. They compare the available photo current with the cathode current as a function of time. In this case, the cathode current lags the source transient by a fraction of an nanosecond.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

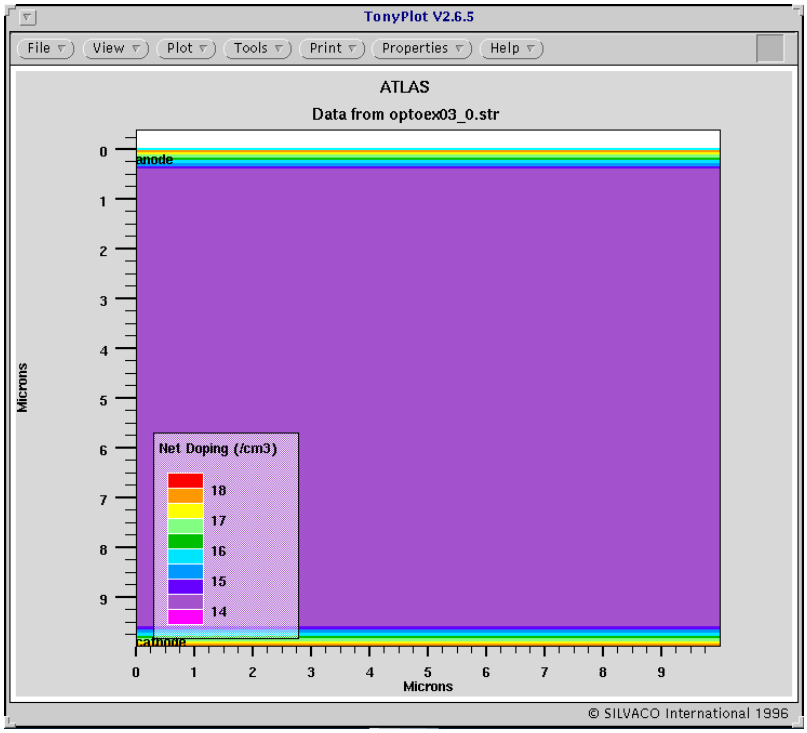


Figure 16.3: Doping profile of the pin detector

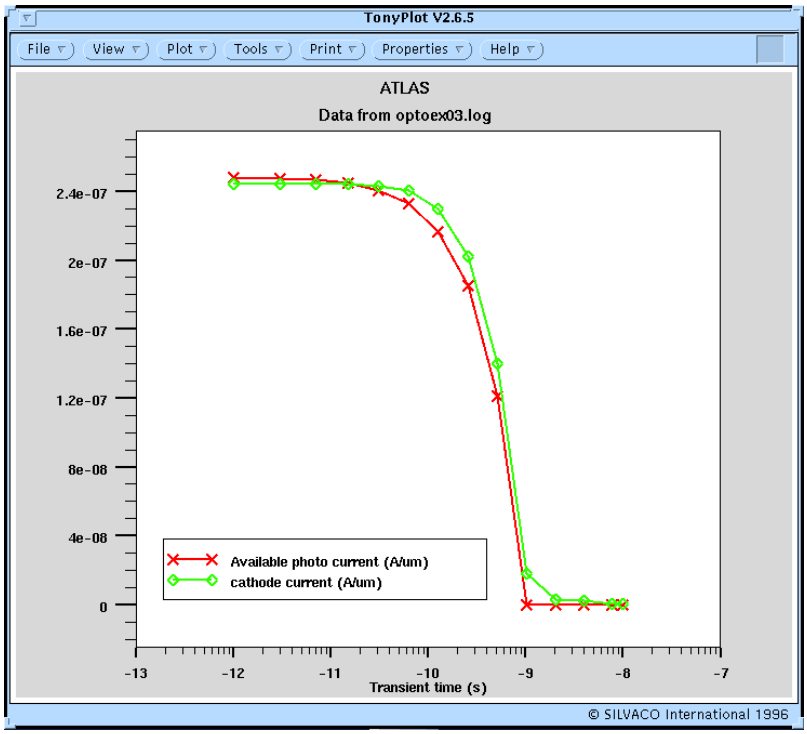


Figure 16.4: Transient simulation of light turn-off. Current output persists for a short time after the light power is zero at 1ns

Input File optoelectronics/optoex03.in:

```
1   go atlas
2   Title Light Turn-Off Transient for PN diode
3   #
4   # Time Domain Photogeneration Simulation
5   # SILVACO International 1992, 1993, 1994, 1995
6   #
7   # SECTION 1: Mesh Specification
8   #
9   mesh space.mult=4.0
10  #
11  x.mesh loc=0.0  spacing=0.25
12  x.mesh loc=10.0 spacing=0.25
13  #
14  y.mesh loc=0.0   spacing=0.05
15  y.mesh loc=5.0   spacing=0.2
16  y.mesh loc=10.0  spacing=0.05
17  #
18  # SECTION 2: Structure Specification
19  #
20  region num=1 material=Silicon
21  #
22  elec num=1 name=anode x.min=0.0 x.max=10.0 y.max=0.0
23  elec num=2 name=cathode bottom
24  #
25  doping uniform conc=1e14 n.type
26  doping gaus peak=0.0 char=0.1 conc=1e18 p.type dir=y
27  doping gaus peak=10.0 char=0.1 conc=1e18 n.type dir=y
28  #
29  # SECTION 3: Material Model Specification
30  #
31  material taup0=2.e-6 taun0=2.e-6
32  models srh auger conmob fldmob
33  #
34  # SECTION 4: Initial Solution
35  #
36
37
38  solve    init outf=optoex03_0.str master
39
40  tonyplot optoex03_0.str -set optoex03_0.set
41
42  #
```

```
# SECTION 5: Optical source specification
#
beam     num=1 x.origin=5.0 y.origin=-1.0 angle=90.0 wavelength=.623
#

method  newton trap
solve    init
solve vcathode=0.1
solve vcathode=0.5
solve vcathode=1.0
solve vcathode=2.0
#
# SECTION 6: Solve dc with light
# ramp the light off in the time domain
# light turns off in 1ns then takes several ns to settle out
#
solve b1=5
log     outf=optoex03.log master
solve b1=0 ramp.lit ramptime=1e-9 tstop=10e-9 tstep=5e-11

tonyplot optoex03.log -set optoex03_1.set
quit
```

16.1.4 optoex04.in: Photodetector : AC Response

Requires: S-PISCES/LUMINOUS

This input file analyzes the small signal response of the device to a small signal optical source.

The first four sections of the input file are identical to the corresponding sections in the input file for the DC example described previously.

The fifth section of the file specifies that a DC solution at a source intensity of 5 W/cm^2 is obtained. After the DC solution, a small signal response of the device to a small signal optical source of 1 mW/cm^2 is extracted at frequencies each decade between the frequencies 100 kHz and 100 GHz. The full AC terminal characteristics are saved in the log file.

When the results are viewed in TONYPLOT it shows that the device has a cutoff frequency of about 1 GHz.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

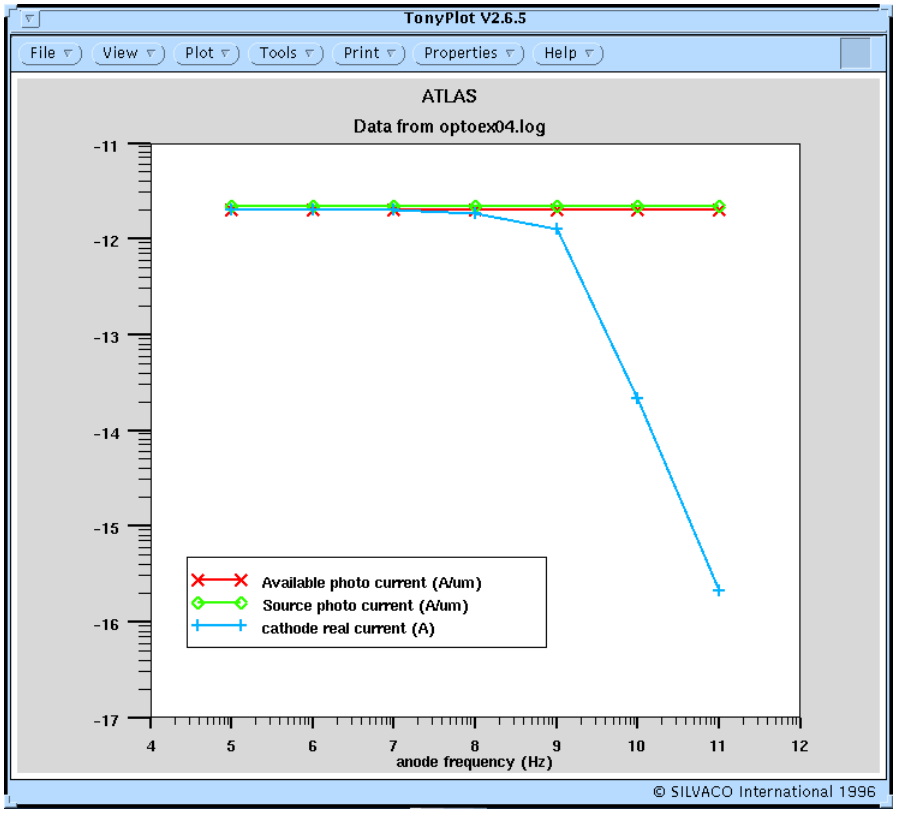


Figure 16.5: AC signal response of the photodetector. The output current falls off at high frequencies of light oscillation.

Input File optoelectronics/optoex04.in:

```
1   go atlas
2   Title PIN diode small signal AC
3   #
4   # Small Signal AC Illumination
5   # SILVACO International 1992, 1993, 1994, 1995
6   #
7   # PIN device description and initial solution
8   #
9   # SECTION 1: Mesh Specification
10  #
11  mesh space.mult=4.0
12  #
13  x.mesh loc=0.0  spacing=0.25
14  x.mesh loc=10.0 spacing=0.25
15  #
16  y.mesh loc=0.0   spacing=0.05
17  y.mesh loc=5.0   spacing=0.2
18  y.mesh loc=10.0  spacing=0.05
19  #
```

```
# SECTION 2: Structure Specification
#
region num=1 material=Silicon
#
elec num=1 name=anode x.min=0.0 x.max=10.0 y.max=0.0
elec num=2 name=cathode bottom
#
doping uniform conc=1e14 n.type
doping gaus peak=0.0 char=0.1 conc=1e18 p.type dir=y
doping gaus peak=10.0 char=0.1 conc=1e18 n.type dir=y
#
# SECTION 3: Material Model Specification
#
material taup0=2.e-6 taun0=2.e-6
models srh auger conmob fldmob
#
# SECTION 4: Initial Solution
#


solve    init outf=optoex04_0.str master

tonyplot optoex04_0.str -set optoex04_0.set

#
# SECTION 5: Optical source specification
#
# define a multi-spectral beam normal to top (y=0.0) surface
# as follows:
# beam #1, origin at (2.5,-1.0), directed at an angle of 90 degrees,
# the beam spectrum is in the file optoex04.spec, the spectrum is
# sampled from 0.5 microns to 0.88 microns with 4 samples.
#
beam     num=1 x.origin=5.0 y.origin=-1.0 angle=90.0 pow-
  er.file=optoex04.spec wavel.start=0.5 wavel.end=0.8 wavel.num=5
#

method  newton trap
solve init
solve    vcathode=0.1
solve    vcathode=0.5
solve    vcathode=1.0
solve    vcathode=2.0
```

```
#
# SECTION 6: AC response
#
log     outf=optoex04.log master
solve b1=.001 ss.phot ss.light=0.001 beam=1 freq=1e5 fstep=10 mult.f nf-
  step=6

tonyplot optoex04.log -set optoex04_1.set

quit
```

16.1.5 optoex05.in: Photodetector: Spectral Response

The final example in this set involves the extraction of spectral response data.

The first four sections of the input file are the same as those in the previous examples. The exception is that a monochromatic wavelength of 100nm is specified. In the last section of the file, solutions are obtained for a 1 W/cm^2 intensity at discrete source wavelengths of 0.1 to 0.8 microns. The `lambda` parameter of the `solve` statement is used to set the wavelength for each solution. The terminal characteristics and wavelength results are saved to the log file.

The results of the simulation shows that the device exhibits good quantum efficiency below a wavelength of about 700 nm.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

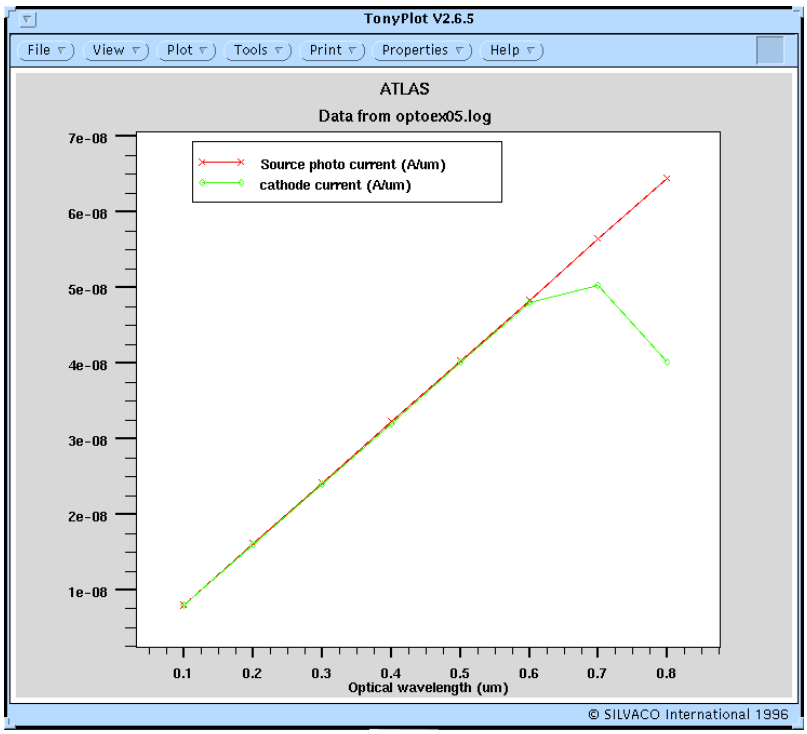


Figure 16.6: Spectral response of the photodetector. Current output falls off at high wavelengths

Input File optoelectronics/optoex05.in:

```
1   go atlas
2   Title PIN photodiode spectral response
3   #
4   # PIN device spectral response.
5   # SILVACO International 1992, 1993, 1994, 1995
6   #
7
8   # SECTION 1: Mesh Specification
9   #
10  mesh space.mult=4.0
11  #
12  x.mesh loc=0.0  spacing=0.25
13  x.mesh loc=10.0 spacing=0.25
14  #
15  y.mesh loc=0.0   spacing=0.05
16  y.mesh loc=5.0   spacing=0.2
17  y.mesh loc=10.0  spacing=0.05
18  #
19  # SECTION 2: Structure Specification
20  #
21  region num=1 material=Silicon
22  #
23  elec num=1 name=anode x.min=0.0 x.max=10.0 y.max=0.0
24  elec num=2 name=cathode bottom
25  #
26  doping uniform conc=1e14 n.type
27  doping gaus peak=0.0 char=0.1 conc=1e18 p.type dir=y
28  doping gaus peak=10.0 char=0.1 conc=1e18 n.type dir=y
29  #
30  # SECTION 3: Material Model Specification
31  #
32  material taup0=2.e-6 taun0=2.e-6
33  models srh auger conmob fldmob
34  #
35  # SECTION 4: Initial Solution
36  #
37
38
39  solve    init outf=optoex05.str master
40
41  tonyplot optoex05.str -set optoex05_0.set
42
```

```
43
44
45 #
46 # SECTION 5: Optical source specification
47 #
48 beam     num=1 x.origin=5.0 y.origin=-1.0 angle=90.0 wavelength=.1
49 #
50
51 method newton trap
52 solve    init
53 solve    vcathode=0.2
54 #
55 # SECTION 5: Spectral Response
56 #
57 log outf=optoex05.log master
58 solve prev b1=1 lambda=0.1
59 solve prev b1=1 lambda=0.2
60 solve prev b1=1 lambda=0.3
61 solve prev b1=1 lambda=0.4
62 solve prev b1=1 lambda=0.5
63 solve prev b1=1 lambda=0.6
64 solve prev b1=1 lambda=0.7
65 solve prev b1=1 lambda=0.8
66
67 tonyplot  optoex05.log -set optoex05_1.set
68 quit
```

16.1.6 optoex06.in: Luminous Efficiency of a III-V LED Device

Requires: BLAZE/LUMINOUS

This example demonstrates how the radiative recombination models in ATLAS can be used to analyze the luminous efficiency and of LED devices.

The objective of this example is to forward bias the led such that radiative recombination occurs in the active layer of the device. Then by extracting the integrated radiative recombination rate and total integrated recombination rate, the luminous efficiency of the device can be measured. Also, the integrated radiative recombination rate as a function of bias voltage can be extracted to estimate the luminous power versus bias voltage.

Mesh Generation

In the first section of the input file a mesh covering an area of 4 microns by 4 microns is defined. The mesh uses the `diag.flip` option since it is symmetric about the center in “X”. Mesh spacings are specified to try to resolve the heterojunctions in the “Y” direction and the doping profiles in the “X” direction. An eliminate statement is used to reduce the number of grid points used in the direction of the substrate. Also, the value of the `space.mult` parameter has been set to 2.0 to provide faster simulation. For better accuracy (and longer simulation time) this value should be set to a smaller number.

Region and Electrode Specification

The device is composed of five regions: a GaAs "p" contact region, an AlGaAs cladding layer, the GaAs active region, a second AlGaAs cladding layer and an n-type GaAs substrate. In this example the grad parameters were used to grade the heterojunctions between the first and second layers and the fourth and fifth layers. The heterojunctions at both sides of the active layer are modeled as abrupt.

A "p" contact electrode is defined at the center of the top of the device. The "p" contact is narrower than the device to confine the photo-luminescence to a narrow region in the active layer. The substrate electrode extends across the device at the bottom.

Doping Profile Specification

The doping profiles in the device are specified in the third section of the input file. Here the first three layers are lightly doped "P" type, while the last two layers are heavily doped "N" type. In addition a "P+" type profile is defined around the "p" contact.

Material Models

In the fourth section of the file, a set of material and model parameters is specified. For analyzing luminous efficiency, all principal recombination mechanisms are selected. In addition, a radiative recombination constant for all the materials is set. In the solution part of the file the individual contributions of the various mechanisms to estimate luminous efficiency will be extracted

Output

In the fifth section of the file, various output parameters that will be written to the solution file during the solution process are chosen. These parameters will allow the user to look at the conduction band edge energy, the valence band edge energy, the total recombination rate, the SRH recombination rate, the Auger recombination rate and the radiative recombination rate.

Initial Solution

In the sixth section of the file, an initial solution with no carriers is done. This usually provides a good initial guess for the full, two-carrier solutions to follow.

Bias Ramp

In the seventh section of the file the bias voltage across the LED is ramped from 0.0 V to 2.0 V in the forward bias direction. In this section a full two carriers solution will be done so that recombination data can be observed. In the first line of this section of the file, an output log file is specified. Current-voltage data will be written to this file. In the solve statement a luminous wavelength The l.wave parameter is set so that luminous power can be extracted during the ramp. In addition, the saving of an output file is set in the solve statement so that snapshots of the device can be displayed at each step of the bias ramp.

Extraction

In the last section of the input file, the integrated total and radiative components of the recombination rate in the device are extracted. The luminous efficiency can be estimated by the ratio of these values.

Results

From these results, the luminous efficiency can be calculated. The luminous efficiency is defined at the ratio of the radiative recombination rate to the total recombination rate. The extracted val-

ues of these rates should be printed at the end of the run. Taking the ratio at a value of around 70% is seen.

To view the results, first start up TONYPLOT with the solution file. This file corresponds to the device structure after the completed voltage ramp. Once TONYPLOT has started, you should display a contour plot of the radiative recombination rate. In this plot we can see the relative luminescence in the active layer compared to other regions of the device. In this case, most of the radiative recombination should be confined to the active layer of the device.

The log file contains all the terminal characteristics. This file contains values of the calculated luminescent power as a function of device bias voltage. Once the file is loaded into TONYPLOT you should select “anode bias” along the x axis and “Luminescent power” along the y axis. In this figure it is seen that the device “turns-on” at about 1.4 Volts.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

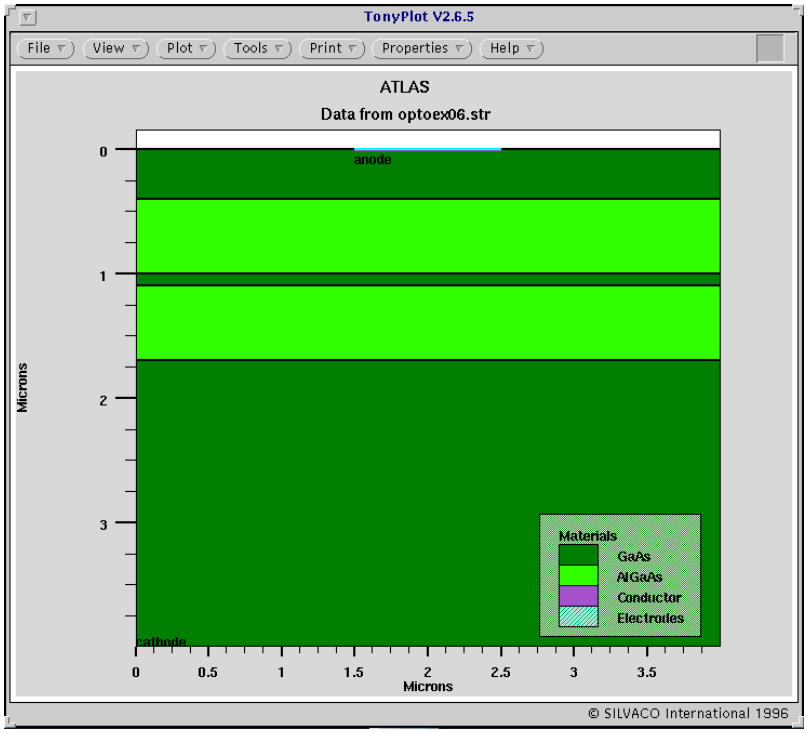


Figure 16.7: Layer Structure of AlGaAs/GaAs LED

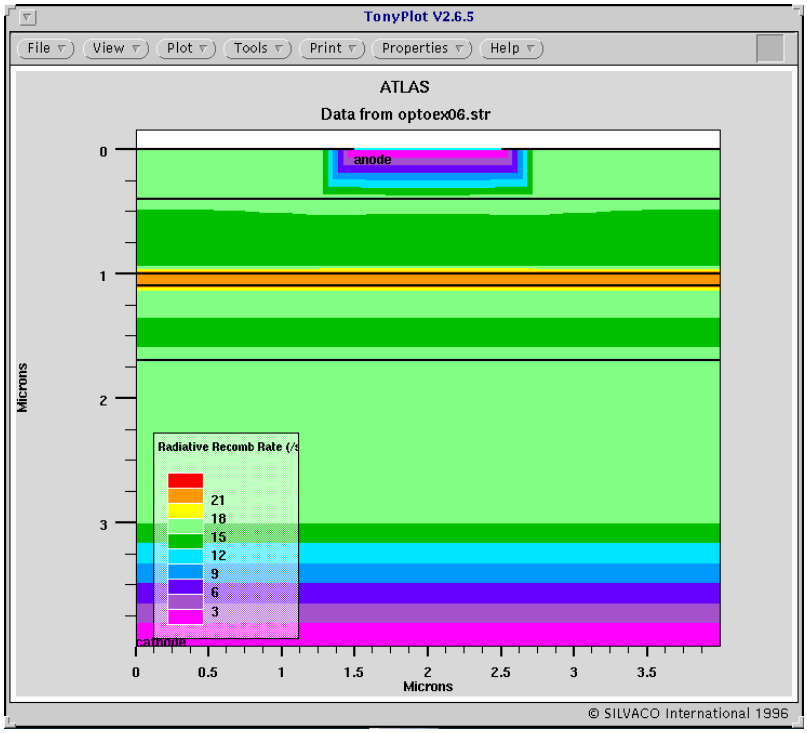


Figure 16.8: Contours of radiative recombination in the LED. The carriers recombine to form light in the thin active GaAs region

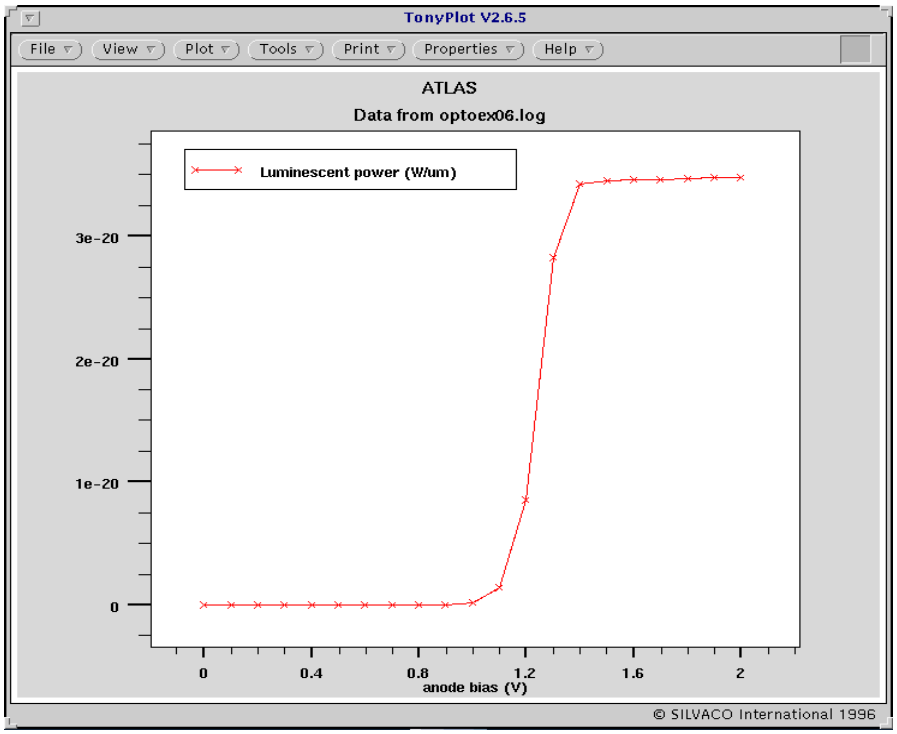


Figure 16.9: Output of light versus applied forward voltage in the LED

Input File optoelectronics/optoex06.in:

```
go atlas
title LED simulation
#
# Simulates a simple LED device
# SILVACO International 1992, 1993, 1994, 1995
#
#-------------------------------------------------------------
# SECTION 1: MESH GENERATION
#-------------------------------------------------------------
# 4 um X 4 um
#
mesh smooth=1 space.mult=2.0
x.mesh l=0.0  spacing=0.25
x.mesh l=1.5  spacing=0.1
x.mesh l=2.5  spacing=0.1
x.mesh l=4.0  spacing=0.25
#
y.mesh l=0.0  spacing=0.2
y.mesh l=0.4  spacing=0.1
y.mesh l=1.0  spacing=0.025
y.mesh l=1.1  spacing=0.025
y.mesh l=1.7  spacing=0.1
y.mesh l=4.0  spacing=0.5
#
eliminate y.direction x.min=0.0 x.max=4.0 y.min=3.0 y.max=4.0
#
#-------------------------------------------------------------
# SECTION 2: REGIONS AND ELECTRODES
#-------------------------------------------------------------
# (graded heterojunctions outside of cladding layers)
#
region num=1 Material=GaAs   y.max=0.5
region num=2 Material=AlGaAs y.min=0.5 y.max=1.0 x.comp=0.35 grad.12=0.1
region num=3 Material=GaAs   y.min=1.0 y.max=1.1
region num=4 Material=GaAs   y.min=1.6
region num=5 Material=AlGaAs y.min=1.1 y.max=1.6 x.comp=0.35 grad.34=0.1
#
elec       num=1  name=anode x.min=1.5 x.max=2.5 y.min=0.0 y.max=0.0
elec       num=2  name=cathode bot
#
#-------------------------------------------------------------
# SECTION 3: DOPING PROFILES
```

```
43 #-------------------------------------------------------------
44 #
45 doping conc=5.0e18 p.type x.left=1.5 x.right=2.5 gaus char=0.5 ra-
     tio.lat=0.6
46 doping uniform region=1 p.type conc=1.e15
47 doping uniform region=2 p.type conc=1.e15
48 doping uniform region=3 p.type conc=1.e15
49 doping uniform region=4 n.type conc=2.e18
50 doping uniform region=5 n.type conc=2.e18
51 #
52 #
53 #-----------------------------------------------------------------
54 # SECTION 4: MATERIAL MODELS
55 #-----------------------------------------------------------------
56 #
57 # (radiative recombination constants same for AlGaAs and GaAs )
58 #
59 material copt=1.5e-10
60 model auger optr srh fldmob bgn fermi
61 #
62 #-----------------------------------------------------------------
63 # SECTION 5: OUTPUT FLAGS
64 #-----------------------------------------------------------------
65 #
66 output con.band val.band recomb u.srh u.aug u.rad
67 #
68 #-----------------------------------------------------------------
69 # SECTION 6: INITIAL SOLUTION
70 #-----------------------------------------------------------------
71 #
72 # no carriers
73
74 method itlim=40
75 solve init
76 #
77 #-----------------------------------------------------------------
78 # SECTION 7: BIAS RAMP
79 #-----------------------------------------------------------------
80 #
81 # 0.0 to 2.0 V forward bias
82 #
83 log outf=optoex06.log master
84 method newton trap autonr
```

```
85  solve  l.wave=0.8 vanode=0.0 vstep=0.1 vfinal=2.0 name=anode
86  save outfile=optoex06.str
87  #
88  # structure plot
89  tonyplot optoex06.str -set optoex06_0.set
90  # radiative recombination rate contours
91  tonyplot  optoex06.str -set optoex06_1.set
92  # luminous intensity vs bias
93  tonyplot optoex06.log -set optoex06_2.set
94  #
95  #-------------------------------------------------------------
96  # SECTION 8: EXTRACTION
97  #-------------------------------------------------------------
98  #
99  # Extract total and radiative components of recombination
100 # (can be used to calculate luminous efficiency)
101 #
102 measure u.total
103 measure u.radiative
104
105 quit
106
```

16.1.7 optoex07.in: User-defined Photo-Generation Rate

Requires: S-PISCES/LUMINOUS/C-INTERPRETER

This example demonstrates how an interpreter function can be used to specify arbitrary generation rate profiles in the device. It shows:

- definition of a 2D diode structure using ATLAS syntax
- setting of a user-defined generate rate in the **beam** statement

In this example, the interpreter functions are used to emulate the absorption of normally incident light. Since this model is a built-in feature of LUMINOUS, the results can be compared with the built-in model. To do this, run `radiate.in` and look at the output. Edit `radiate.in` and comment out the parameter, `f.radiate=optoex07.lib`, then run the file again. In either case the results should match. The function in `optoex07.lib` can be varied to prove that the interpreted functions are being used in the first case.

The user-defined function is in the text file `optoex07.lib`. This file is copied to the current working directory when you load this example. This file can be edited manually to set any function or shape of photogeneration profile in the device. The value of the photogeneration rate from the subroutine is multiplied by the value of the `b1` parameter in the `solve` statement.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

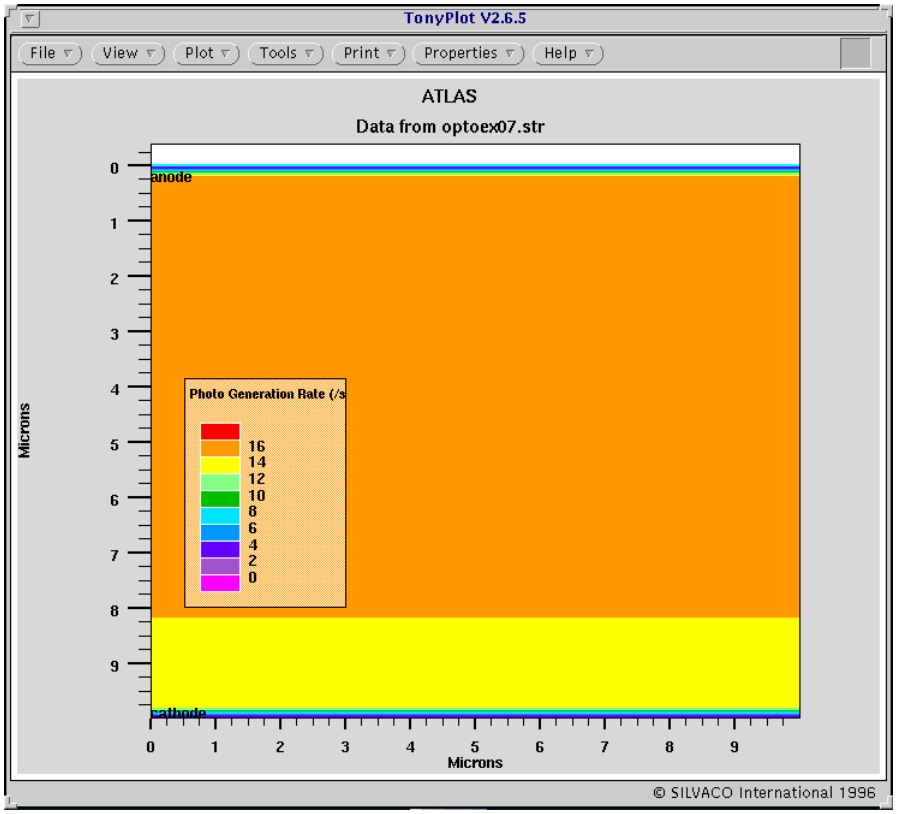


Figure 16.10: Photogeneration contours from a user-defined generation function

```
/*
 * ------------------------------------------------------------------
 * This function can be used to describe arbitrary generation rate versus
 * location.
 * ------------------------------------------------------------------
 */
int radiate(x,y,t,g)
double x;    /* device x coordinate (microns) */
double y;    /* device y coordinate (microns) */
double t;    /* simulation time (s) */
double *g;   /* generation rate (pairs/sec/cubic cm) */
{
        double alpha, dum;
        alpha = 5219.459468;
        dum = 0.6019e-6/(3.0e8*6.626e-34)*alpha*exp(0.0-alpha*y*1e-4);
        *g = dum;
        return(0);
}
```

Figure 16.11: C-language function used to describe photogeneration in this example. The function is interpreted at run-time by ATLAS

Input File optoelectronics/optoex07.in:

```
1  go atlas
2  Title Radiation interpreter function example
3  #
4  # Radiation interpreter function example
5  # SILVACO International 1993, 1994, 1995
6  #
7  # SECTION 1: Mesh Specification
8  #
```

```
mesh space.mult=4.0
#
x.mesh loc=0.0  spacing=0.25
x.mesh loc=10.0 spacing=0.25
#
y.mesh loc=0.0   spacing=0.05
y.mesh loc=5.0   spacing=0.2
y.mesh loc=10.0  spacing=0.05
#
# SECTION 2: Structure Specification
#
region num=1 material=Silicon
#
elec num=1 name=anode x.min=0.0 x.max=10.0 y.max=0.0
elec num=2 name=cathode bottom
#
doping uniform conc=1e14 n.type
doping gaus peak=0.0 char=0.1 conc=1e18 p.type dir=y
doping gaus peak=10.0 char=0.1 conc=1e18 n.type dir=y
#
# SECTION 3: Material Model Specification
#
material taup0=2.e-6 taun0=2.e-6
models srh auger conmob fldmob
#
# To run the built in model comment out the line with f.radiate in it.
# The results should match.
#
beam     num=1 x.origin=5.0 y.origin=-1.0 angle=90.0 \
         wavelength=0.6019  f.radiate=optoex07.lib
#
# SECTION 4: Initial Solution
#


solve init
#
# SECTION 5: Current vs. intensity
#
method newton
solve b1=0.000001 outf=optoex07.str master
tonyplot optoex07.str -set optoex07.set
quit
```

16.1.8 optoex08.in: Solar Cell Simulation

Requires: SSUPREM4/S-PISCES/LUMINOUS

This example shows how a solar cell structure can be simulated in ATHENA and ATLAS. The structure used and the results obtained are similar to those in the book "VLSI Technology" by Sze. The example file consists of:

- construction of solar cell doping and geometry in ATHENA
- simulation of short circuit current
- simulation of open circuit voltage
- simulation of spectral response

The initial part of the example file uses ATHENA to create a typical solar cell structure. The device used is a diode of n+ over a p substrate. The junction depth is approx 0.25um. A single contact is placed in the center of the structure. Typical solar cells will have this structure repeated many times across a large area. Here advantage is taken of the reflecting boundary conditions at the edges of the structure to simulate just one cathode contact.

The process consists of an implant and diffusion followed by electrode formation. The electrode statement is then used to define the electrodes for ATLAS.

The first ATLAS run in the file simulates the response of the device to illumination by the solar spectrum. At the start of the file the material parameters for the structure are set. The imaginary refractive index (which is directly related to the absorption coefficient) of the metal is defined to a high value to ensure it is opaque to the incident radiation.

The illumination by solar spectrum is defined by the `beam` statement. The origin and angle of incidence need to be defined. An angle of 90 degrees means normal incidence from the top. The parameter, `power.file`, points to an external file that contains a list of wavelength vs. intensity. The file, `optoex08.spec`, contains data for the Air Mass Zero Solar Spectrum. The intensity in the file can be considered just as relative intensity between the wavelengths. The actual power in Watts/cm^2 is set by the `b1` parameter of the `solve` statement.

It is possible to store the optical intensity of the illumination by specifying `output opt.int` at any time before saving a structure file. The photogeneration rate will appear in the solution structure file by default.

The short circuit current is the current when anode and cathode are shorted. This is simulated by illuminating the device with zero voltage on all contacts. The `B<n>` parameter of the `solve` statement sets the power of the light defined as beam `<n>`. An `extract` statement is used to measure the short circuit current. Extracted target like this can be used for optimization or as targets in the VWF.

The system is reset using the command 'solve init' for the second test. To simulate open circuit voltage, the current through the device is forced to be zero. The device is illuminated and the voltage sustained across the device is measured.

To set the current to zero, it is first necessary to set one electrode to have current boundary conditions. This is done on the cathode using the `current` parameter of the `contact` statement. Simulating with the device illuminated and `icathode=0` gives the open circuit voltage. This voltage is measured using the `extract` statement. Note the use of the syntax, `vint. "cathode"`, as opposed to `v. "cathode"`. This is necessary when you want to extract the voltage directly on the semiconductor contact as opposed to the applied bias.

The structure stored at this point can be plotted to show contours of photogeneration, carrier concentration or potential. The photogeneration contours shows the effects of the opaque cathode contact. It is also possible to show ray tracing of light beams in this file. However, in our example, normal illumination was used so there are no refractive effects to see. A similar run using non-normal incidence and a narrower illumination spot would show the ray tracing inside the device.

The second ATLAS run in this file is used to simulate the spectral response of the solar cell. At the start the definition of material parameters and opaque metal contact is repeated. For this test a different light beam is required. A beam origin and angle of incidence are set as before. The wavelength or spectrum file is not defined since the wavelength is to be varied in this test.

The short circuit case is considered so zero biases are set on both electrodes. The parameter `lambda` of the `solve` statement is used to set the wavelength of the incident light. The range of wavelength used was 300nm to 1000nm.

Plotting the resulting `log` file, it is possible to see how the cathode current varies with wavelength. Setting a plot vs wavelength of source photocurrent (current available in the light beam), available photocurrent (current available for collection) and actual cathode current can show how the device behaves. The losses between source and available photocurrent are caused by reflections and transmission. This dominates the losses at all except the shortest wavelengths. The losses from available photocurrent to the actual simulated cathode current are due to recombination. These are very low except at the extremes.

This simulation has produced all the data needed to create a plot of efficiency vs light frequency as used in the reference. All that is needed is to transform the data in TONYPLOT. By using TonyPlot functions, the user can set the x-axis to be light energy by:

```
Light energy in eV = hc/q(lambda)
```

where h=Planck's constant, c=speed of light, and q=electronic charge. In TONYPLOT the function needed is :

```
6.6e-34*3.0e+8/(1.6e-19*Optical Wavelength)
```

On the y-axis the efficiency is needed. This is defined as (power in/power out). In TONYPLOT the following function is needed:

```
Source Photocurrent/cathode current.
```

This plot shows the same functional form as the reference material. Peak efficiency is around 0.82. It falls off quickly at low energies and more slowly at higher energies.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

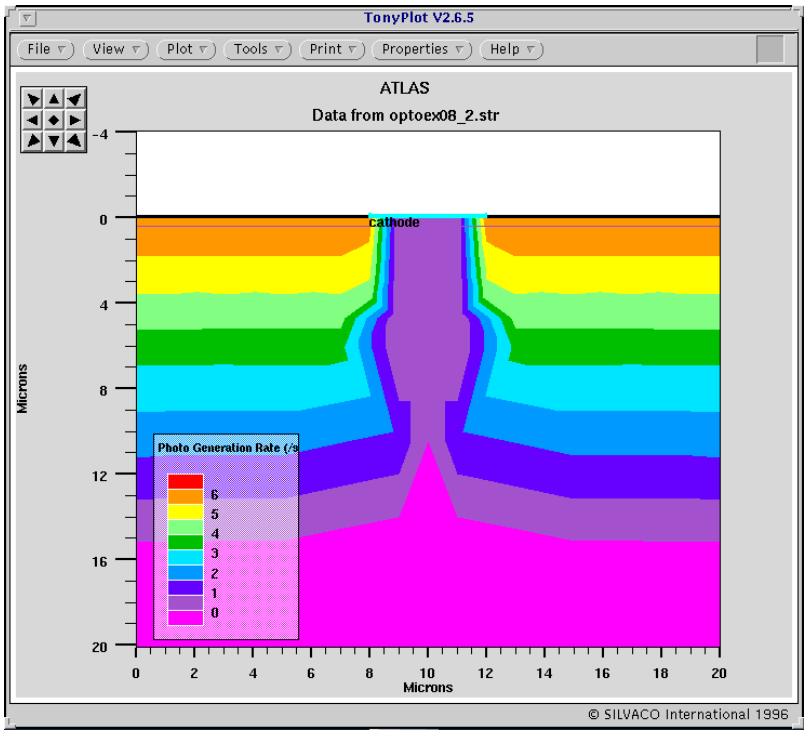


Figure 16.12: Photogeneration contours in a 2D Solar Cell. The center cathode electrode is opaque

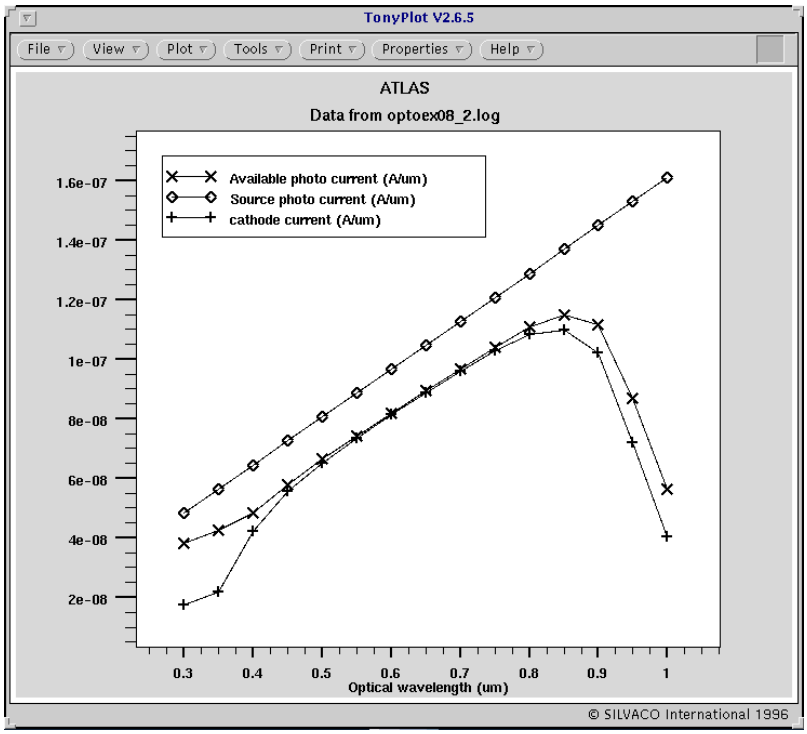


Figure 16.13: Spectral response of the solar cell shows fall off in efficiency at high and low wavelengths

Input File optoelectronics/optoex08.in:

```
1    go athena
```

```
#
line x loc=0.00 spac=1
line x loc=10 spac=1
#
line y loc=0.00 spac=0.05
line y loc=0.25 spac=0.02
line y loc=1 spac=0.1
line y loc=50 spac=10


init silicon c.boron=1.0e14 orientation=100

# deposit oxide coating
deposit oxide thickness=0.05

# implant n+ layer
implant phos dose=1e15 energy=30

# drive-in
diffuse time=10 temp=900

# extract n layer junction depth
extract name="junc_depth" xj material="Silicon" mat.occno=1 x.val=0.1
  junc.occno=1

# form contact
etch oxide right p1.x=8
deposit alum thickness=0.1 div=3
etch alum left p1.x=8

# relax the mesh in deep area
relax y.min=0.6
relax y.min=2.0
relax y.min=10

# Reflect to get complete structure
structure mirror right

# set electrodes for ATLAS
electrode name=cathode x=10
electrode name=anode backside

structure outf=optoex08_0.str
```

```
44
45 go atlas
46
47 # set contact material to be opaque
48 material material=Aluminum imag.index=1000
49
50 material material=Silicon  taun0=1e-6 taup0=1e-6
51
52
53 # set light beam using solar spectrum from external file
54 beam  num=1 x.origin=10.0 y.origin=-2.0 angle=90.0 pow-
     er.file=optoex08.spec
55
56 # saves optical intensity to solution files
57 output opt.int
58
59 models conmob fldmob srh print
60
61 solve
62
63 # get short circuit current
64 log outf=optoex08_0.log
65 solve b1=1.0e-15
66
67 extract name="short_circuit_current" max(abs(i."cathode"))
68 save outf=optoex08_1.str
69
70 # get open circuit voltage
71 solve init
72 contact name=cathode current
73 solve icathode=0 b1=1e-15
74
75 extract name="open_circuit_voltage" max(abs(vint."cathode"))
76 save outf=optoex08_2.str
77 tonyplot optoex08_2.str -set optoex08_2.set
78
79
80 go atlas
81 #
82 # SECOND ATLAS RUN FOR SPECTRAL RESPONSE
83 #
84
85
```

```
# set contact material to be opaque
material material=Aluminum imag.index=1000

material material=Silicon  taun0=1e-6 taup0=1e-6


# set monochromatic light beam for spectral analysis
beam  num=1 x.origin=10.0 y.origin=-2.0 angle=90.0

# saves optical intensity to solution files
output opt.int

models conmob fldmob srh print

# spectral response
solve init b1=0
log outf=optoex08_2.log
solve b1=1 lambda=0.3
solve b1=1 lambda=0.35
solve b1=1 lambda=0.4
solve b1=1 lambda=0.45
solve b1=1 lambda=0.5
solve b1=1 lambda=0.55
solve b1=1 lambda=0.6
solve b1=1 lambda=0.65
solve b1=1 lambda=0.7
solve b1=1 lambda=0.75
solve b1=1 lambda=0.8
solve b1=1 lambda=0.85
solve b1=1 lambda=0.9
solve b1=1 lambda=0.95
solve b1=1 lambda=1.00

tonyplot optoex08_2.log -set optoex08_3.set

quit
```

16.1.9 optoex09.in : Use of Anti-Refective Coating

Requires: SPISCES/LUMINOUS

This example shows how anti-reflective coatings can be simulated in ATLAS. This example demonstrates:

- construction of simple silicon region
- specification of normally incident light beam
- definition of anti-reflective layer using the INTERFACE statement
- simulation of spectral response

The structure and beam definition sections of this example are similar to other simple examples in this section. The wavelength of the beam will be set on the solve lines so it is not defined in the beam statement.

The interface statement is used to define the properties of the anti-reflective layer on the top of the silicon. Note that the layer does not need to be physically present in the structure defined for ATLAS. The parameters ar.thick defines the ARC thickness. Most commonly this is a quarter wavelength. ar.index defines the refractive index of the ARC.

The lambda parameter varies the wavelength of the light beam using the solutions. The refractive indixes in the structure can be checked using the index.check parameter. This produces output of the real and imaginary index in the silicon as a function of the specified wavelength.

The final plot compares the spectral response of the cases with and without the ARC. Note the increase around 0.6um wavelength for the ARC. In this example the terminal current is not important since we consider a one terminal device with no applied bias. The source photocurrent is the amount of current generated by the light source. available photocurrent is the amount of current absorbed by the semiconductor. Differences between these two are due to reflection, transmission or absorption in non-semiconductor materials. The ratio of available/source photocurrents is often known as external quantum efficiency. This quantity can be plotted using the functions in Tonyplot.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

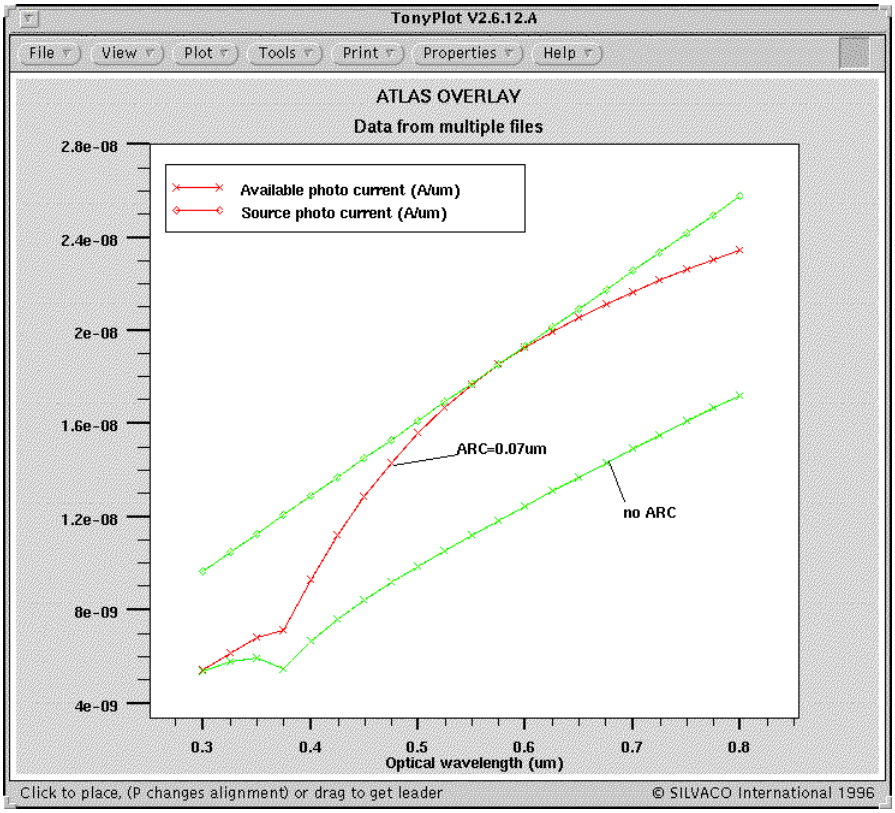


Figure 16.14: Plot of available and source photocurrent with no ARC and with a 0.07um ARC

Input Deck optoelectronics/optoex09.in :

```
1   go atlas
2
3   mesh space.mult=1.0
4   x.mesh loc=0.0  spacing=10.0
5   x.mesh loc=10.0  spacing=10.0
6   y.mesh loc=0.0   spacing=0.2
7   y.mesh loc=50.0  spacing=0.2
8
9   region num=1 material=Silicon
10  #
11  elec  name=cathode bottom
12  #
13  doping uniform conc=1e14 n.type
14
15  # define a beam (be sure to include REFLECT and BACK parameters)
16  beam num=1 x.origin=5.0 y.origin=-1.0 angle=90.0    \
17       back.refl front.refl reflect=5 min.w=-2 max.w=2 \
18       min.power=0.001
19
20  #define anti-reflective coating
21  interface ar.index=2.05 ar.thick=0.07
22  #p1.x=0.0 p1.y=0.0 p2.x=10.0 p2.y=0.0
23
24  log outf=optoex09.log
25
26  solve b1=1 lambda=0.3 index.check
27  solve b1=1 lambda=0.325 index.check
28  solve b1=1 lambda=0.35 index.check
29  solve b1=1 lambda=0.375 index.check
30  solve b1=1 lambda=0.4 index.check
31  solve b1=1 lambda=0.425 index.check
32  solve b1=1 lambda=0.45 index.check
33  solve b1=1 lambda=0.475 index.check
34  solve b1=1 lambda=0.5 index.check
35  solve b1=1 lambda=0.525 index.check
36  solve b1=1 lambda=0.55 index.check
37  solve b1=1 lambda=0.575 index.check
38  solve b1=1 lambda=0.6 index.check
39  solve b1=1 lambda=0.625 index.check
40  solve b1=1 lambda=0.65 index.check
41  solve b1=1 lambda=0.675 index.check
42  solve b1=1 lambda=0.7 index.check
```

```
43 solve b1=1 lambda=0.725 index.check
44 solve b1=1 lambda=0.75 index.check
45 solve b1=1 lambda=0.775 index.check
46 solve b1=1 lambda=0.8 index.check
47
48 tonyplot -overlay optoex09.log optoex09_noarc.log  -set optoex09.set
```

17.1. CCD: CCD Application Examples

17.1.1 ccdex01.in: Illumination and Charge Transfer

Requires: SSUPREM4/S-PISCES/LUMINOUS

In this example a CCD structure is constructed using ATHENA process simulation. This structure is passed to ATLAS for electrical testing. The input file tests consist of four portions:

- construction of a CCD device
- emptying the storage well of carriers
- illuminating the CCD with a light source and measuring the generated carriers
- transfer of the carriers to the drain

The first stage of the input constructs the CCD geometry and doping profiles in ATHENA. The CCD device consists of a storage node on the left of the structure controlled by a polysilicon gate. A transfer node to the right of this also controlled by both a polysilicon gate and a drain region of heavy n+ doping with a metal contact.

The CCD structure has an n- active region at the surface above a p-type substrate. Under the transfer node, an extra p-type implant is given. This implant is off sufficient magnitude to create a potential difference between the storage and transfer nodes, but is not so large as to cause a junction.

The process simulation is designed to run extremely quickly so there is a minimum of diffusion steps included. The final stages of the ATHENA input performs the electrode definition needed by ATLAS. Also some extract statements are used to measure the junction depth of the active region under the storage and transfer nodes.

The ATLAS simulation begins with definition of the models and material parameters of the device. The `contact` statement is used to define the workfunction of the two polysilicon electrodes. The `material` statement is used to define minority carrier lifetimes in the semiconductor. The interface charge between the gate oxide and the semiconductor is set by the `qf` parameter of the `interface` statement.

The physical models used in this simulation reflect the different physical effects important in CCD devices. The mobility model `CVT` includes a surface mobility degradation which is important in the transfer phase of the simulation as the charge moves from the storage node to the drain. The recombination models *consrh* and *auger* are needed as the recombination of the carriers is important in determining the capacity and charge transfer characteristics. Band gap narrowing is also needed to model the properties of areas with high carrier concentration.

The initial part of the ATLAS input file biases the drain to +20V, while the two gates are held at -6V. This is done to deplete the electrons from the n- active layer. The final structure with the active layer empty of carriers is saved. At this point `extract` statements are used to extract the integrated electron concentration in the storage well. This will be a small number.

Next, LUMINOUS is used to illuminate the storage node. The light beam is defined near the start of the input file using the `beam` statement. This beam is defined with a 500nm wavelength, an origin at (0,-1) and width 2um. The angle of 90 degrees specifies normal incidence onto the top of the CCD. LUMINOUS calculates all internal reflections and absorption by the layers withing the simulation structure.

To illuminate the structure, a transient simulation is used. The power of the light beam is controlled by the parameter `B1` on the solve statement. This is defined to rise from zero to 1 W/cm^2 in 5ns. to remain at this power for a further 15ns, and then to fall to zero in another 5ns. The tstep parameter is used to define the initial timestep of the simulation. All other timesteps are automatically calcu-

lated by ATLAS based on the local truncation error. The `tstop` parameter defines the end of the simulation time. The illumination part is set to last 50ns.

At the end of the illumination section, a structure is stored. This structure can be plotted to see the charge collected under the storage node. The potential difference between the storage and transfer nodes created by the extra boron implant stops the charge from leaking away to the large positive potential on the drain. `Extract` statements are used once again to measure the integrated electron concentration. The value obtained is subtracted from the value obtained from the empty well.

The final stage of the ATLAS simulation is the charge transfer. This is also performed as a transient simulation. It is important to note that during a single ATLAS run the time is *not* reset to zero. Since the illumination section was 50ns, that is the starting time for this section. In this section the transfer gate is ramped from -6V to +2V. This causes the potential barrier between the storage node and the drain to be removed. Charge can now flow from the storage node, under the transfer gate and into the drain.

At each timestep in the transfer simulation, a structure is saved. By loading several of these structure files into TONYPLOT, it is possible to observe the movement of the charge across the device.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

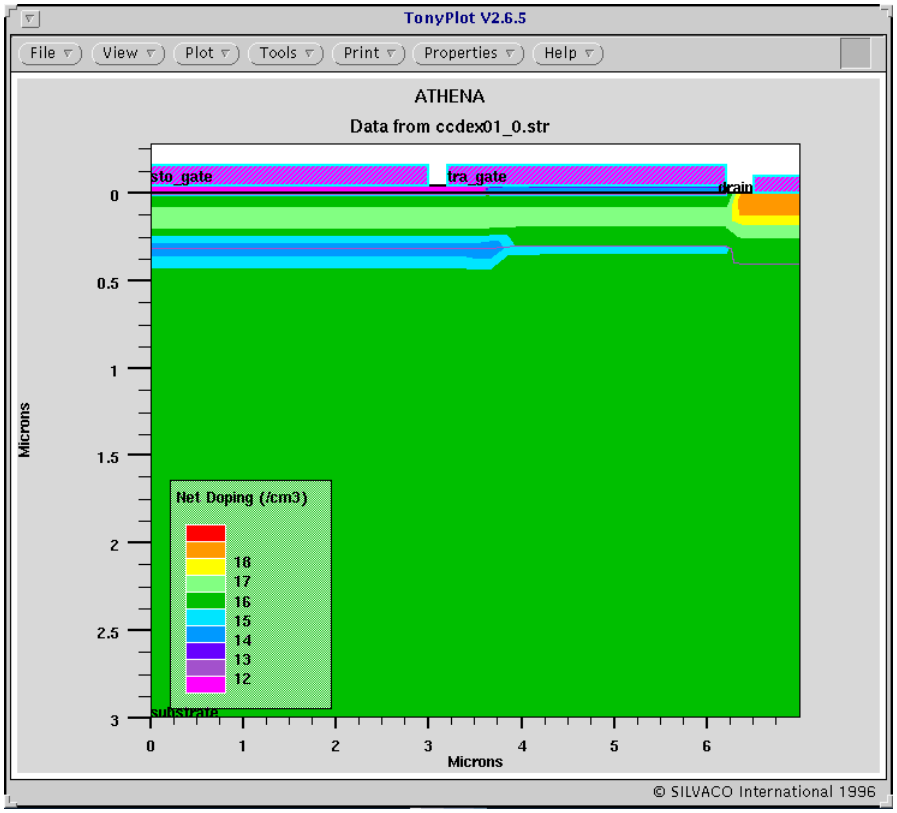


Figure 17.1: Structure and Doping of CCD device with two gates and a drain

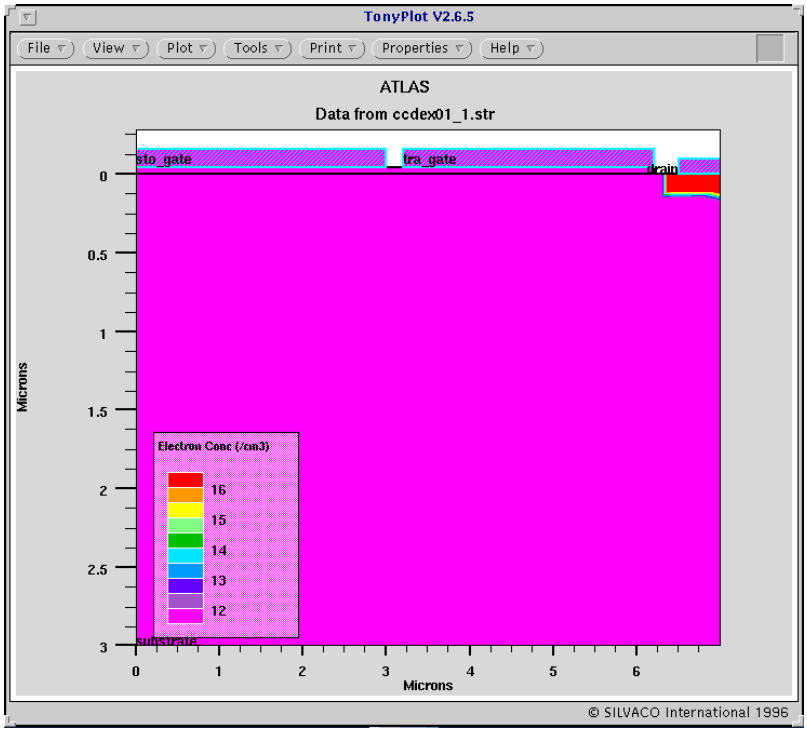


Figure 17.2: Electron concentration after emptying of the channel. Electron concentration is essentially zero under the gates

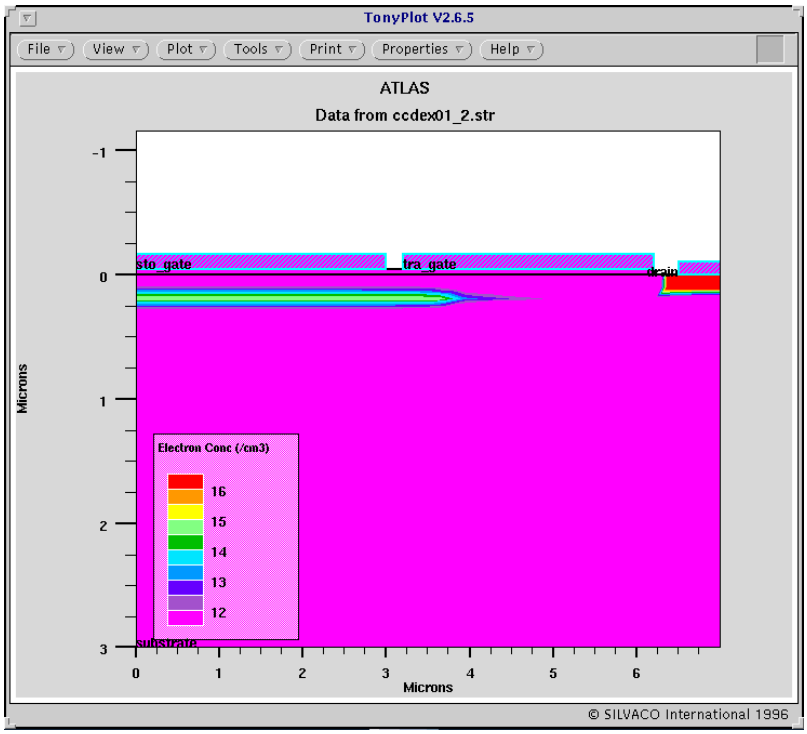


Figure 17.3: Electron concentration after illumination. Charge is stored under STO_GATE

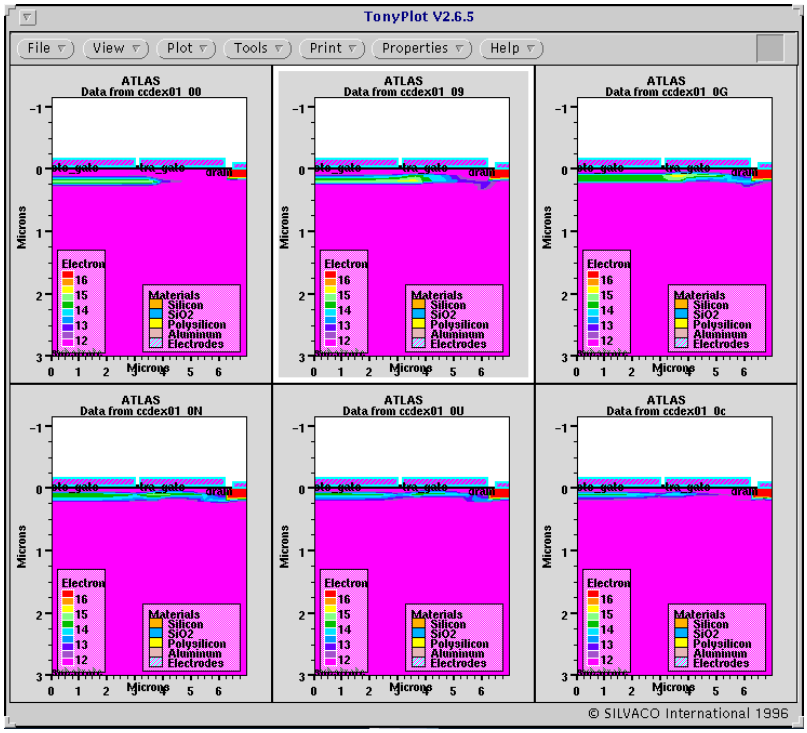


Figure 17.4: Sequence showing electron concentration during charge transfer. A Movie can be made in TONYPLOT to show the moving charge

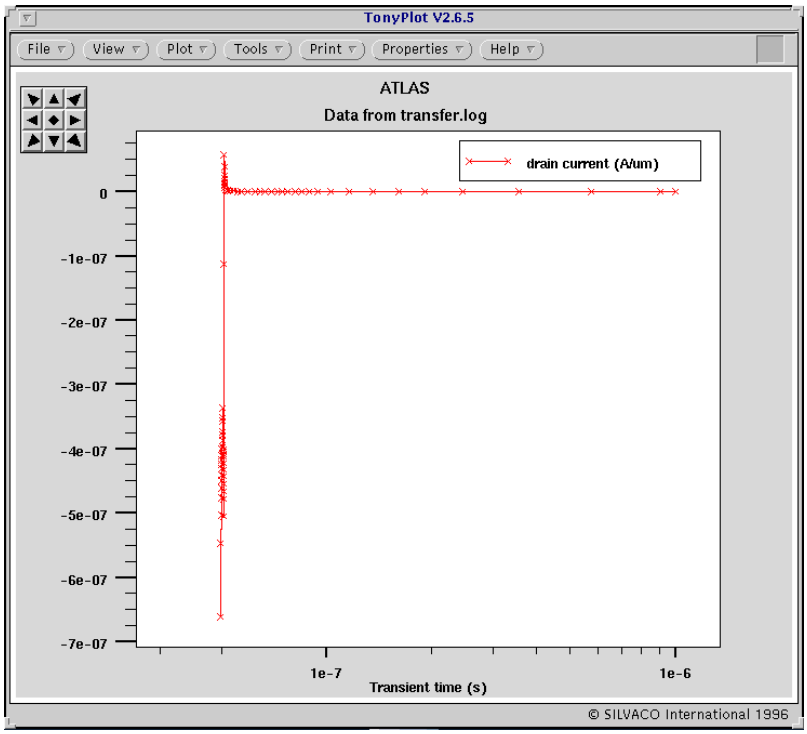


Figure 17.5: Terminal drain current during charge transfer

Input File ccd/ccdex01.in:

```
1   go athena
2
3   line x loc= 0.00 spac=0.5
4   line x loc= 3.00 spac=0.1
5   line x loc= 3.20 spac=0.1
6   line x loc= 4.70 spac=0.25
7   line x loc= 6.20 spac=0.1
8   line x loc= 7.00 spac=0.2
9
10  line y loc=0.00  spac=0.05
11  line y loc=0.6   spac=0.1
12  line y loc=3.0   spac=0.5
13
14
15  # Define material, orientation, initial doping
16  init silicon boron conc=5e15 orientation=100 space=2
17
18  # form n- active layer at surface
19  implant phosphorus dose=8e11 energy=120
20
21  # active layer diffusion
22  diffuse time=60 temp=950 nitro
23
24  # Deposit gate oxide
25
26  deposit oxide thick=0.04 divisions=3
27
28
29  # mask for the p- implant under transfer gate
30  deposit photo thick=3.0 divisions=2
31
32  etch photores start x=7 y=-10.0
33  etch cont x=7 y=10.0
34  etch cont x=3.5 y=10.0
35  etch done x=3.5 y=-10.0
36
37
38  relax y.min=1.0
39
40
41  # Implant p- region
42  implant boron dose=1e11 energy=80
```

```
43
44 etch photores all
45
46 #drive diffusion : commented out for speed
47 #method fermi compress
48 #diffus time=220 temp=1000 nitro press=1.00
49
50 deposit poly thick=0.12
51
52 etch poly start x=3.0 y=-5
53 etch cont x=3.0 y=5
54 etch cont x=3.2 y=5
55 etch done x=3.2 y=-5
56
57 etch poly start x=6.2 y=-5
58 etch cont x=6.2 y=5
59 etch cont x=7.0 y=5
60 etch done x=7.0 y=-5
61
62 deposit photo thick=1.0 divisions=8
63
64 etch photo right p1.x=6.2
65 etch oxide right p1.x=6.2
66
67 # Implant n+ drain region
68 implant arsenic dose=1e15 energy=50
69
70 strip
71
72 #drain anneal : commented out for speed
73 #method fermi compress
74 #diffus time=30 temp=900 nitro press=1.00
75
76 deposit alum thick=0.1 div=2
77 etch alum left p1.x=6.5
78
79 electrode name=sto_gate x=0
80 electrode name=tra_gate x=4
81 electrode name=drain x=6.9
82 electrode name=substrate backside
83
84 # extract process parameters
85 # here active layer/substrate junction is SECOND from the surface
```

```
86  extract name="storage_xj" xj material="Silicon" mat.occno=1 x.val=1
      junc.occno=2
87  extract name="transfer_xj" xj material="Silicon" mat.occno=1 x.val=5
      junc.occno=2
88
89
90
91  structure outf=ccdex01_0.str
92  tonyplot ccdex01_0.str -set ccdex01_0.set
93
94
95  go atlas
96
97  contact name=sto_gate n.poly
98  contact name=tra_gate n.poly
99
100 material taun0=1e-7 taup0=1e-7
101 models cvt consrh bgn auger print
102 interface qf=3e10
103
104 beam num=1 wavelength=0.6 x.ori=0 y.ori=-1 max.win=2 angle=90
105 solve init
106
107 method newton climit=1e-4
108
109 solve v1=-6 v2=-6
110 solve vdrain=0.1
111 solve vdrain=1 vstep=1 vfinal=15 name=drain
112
113 #save empty structure
114 save outf=ccdex01_1.str
115 tonyplot  ccdex01_1.str -set ccdex01_1.set
116
117
118
119
120
121 #extract electron concentration in empty well (1.0e4 is to scale into I/
      cm^3)
122 extract init inf="ccdex01_1.str"
123 extract name="n_in_empty_well" 1.0e+4*area from curve(depth,n.conc mate-
      rial="Silicon" mat.occno=1 x.val=1.0)
124
125 # now turn on light beam in transient
```

```
126
127 solve b1=1 ramp.lit ramptime=5e-9 dt=1e-11 tstop=20e-9 outf=light.str
      master onefile
128 solve b1=0 ramp.lit ramptime=5e-9 dt=1e-10 tstop=50e-9
129
130 #save charged structure
131 save outf=ccdex01_2.str
132 tonyplot  ccdex01_2.str -set ccdex01_2.set
133
134
135
136 #extract electron concentration in full well (1.0e4 is to scale into I/
      cm^3)
137 extract init inf="ccdex01_2.str"
138 extract name="n_in_full_well" 1.0e+4*area from curve(depth,n.conc materi-
      al="Silicon" mat.occno=1 x.val=1.0)
139 extract name="stored_n" $n_in_full_well - $n_in_empty_well
140
141
142 # now do transfer
143 # ramp transfer gate to +2V
144 log outf=transfer.log
145 solve v2=2 ramptime=1e-9 dt=1e-11 tstop=1e-6 outf=ccdex01_00 master
146
147
148 tonyplot ccdex01_00 ccdex01_09 ccdex01_0G ccdex01_0N ccdex01_0U
      ccdex01_0c -set ccdex01_3.set
149
150 quit
151
```

17.1.2 ccdex02.in: Emptying The Storage Well Using Quasi Fermi Levels

Requires: SSUPREM4/S-PISCES/LUMINOUS

In this example a CCD structure is constructed using ATHENA process simulation. This structure is passed to ATLAS for electrical testing. The input file tests consist of three portions:

- construction of a CCD device
- emptying the storage well of carriers by directly setting the quasi-fermi levels
- illuminating the CCD with a light source and measuring the generated carriers

This example differs from the previous one in that **no drain contact is used.** The emptying of the well is done by setting the quasi-fermi level of the n- active region directly.

The structure of this CCD is two polysilicon gates on top of silicon, separated by a nitride/oxide insulator layer. The right hand gate is used to collect the charge and the left hand gate is used to confine this charge under the right-hand gate.

At the start of the input file is the ATHENA process to produce the structure as described for the previous example. Electrodes are defined in ATHENA and `extract` statement which are used to measure the junction depth of the n- active layer.

In the ATLAS simulation the material and models definition is similar to the previous example. The CVT surface transport model is not needed as this simulation does not include charge transfer.

The key technique demonstrated by this example is the use of setting the quasi fermi levels directly in ATLAS. To set the quasi fermi levels for electrons, the parameter `n.bias` is used. For this to work, of course, it is necessary not to solve for the carriers in question. The setting of `method carriers=0` specifies a zero carrier solution. Only Poisson's equation for potential is solved.

Using 'n.bias' the electron quasi fermi level is set to +20V. This is a non equilibrium condition so the only way to use the solution of the continuity equations to observe the electron concentration is in transient mode. ATLAS is switched to solve for electron and holes using `method carriers=2.` Then a short transient simulation is done and a structure file saved. Plotting electron concentration in this structure file shows the CCD storage well to be depleted of electrons. An `extract` statement is used to quantify this.

In this CCD biasing scheme the storage node to the left is biased to+12V and the containment node to the left is biased at -8V. This ensures that charges generated under the storage gate are collected there.

In the final stage of the input file the CCD storage node is illuminated. The light beam is defined in the BEAM statement to be normally incident from the top, with an origin of (6.85,-1), width 4um and wavelength 800nm. All reflections, refraction and absorption in the layers of the structure is simulated by ATLAS.

A transient illumination is defined lasting 1us. At the end of this time the amount of electrons trapped in the CCD can be measured using the `extract` statement.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory.

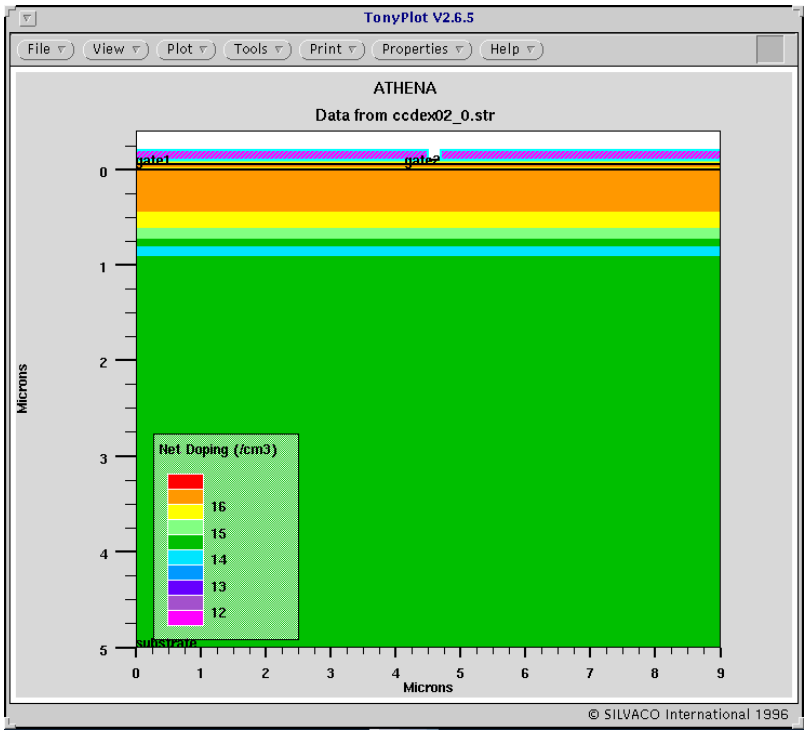


Figure 17.6: Structure and Doping of CCD device with two gates, but without a drain

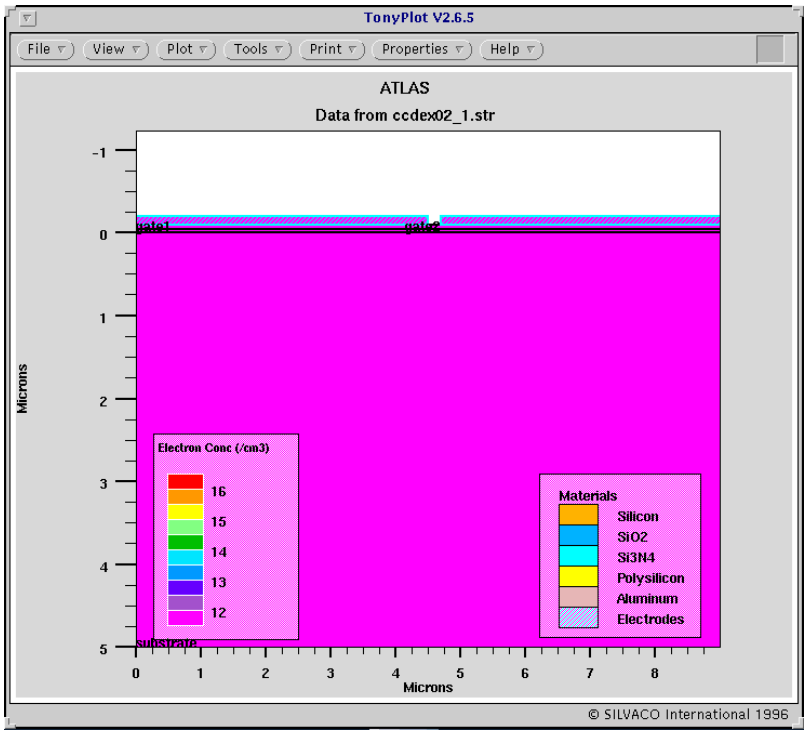


Figure 17.7: Electron Concentration in the empty well state simulated using the QFL method (no electrons present)

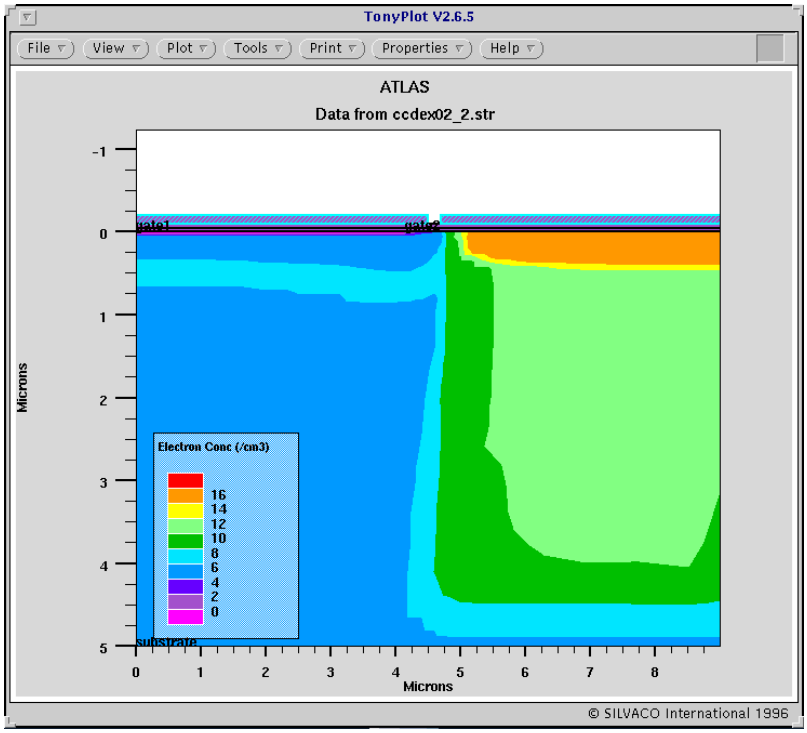


Figure 17.8: Electron concentration after illumination of right hand side of the structure

Input File ccd/ccdex02.in:

```
1   go athena
2   # CCD  structure simulation
3   # SILVACO International 1994
4   #
5   line x loc= 0.00 spac=0.2
6   line x loc= 0.45 spac=0.25
7   line x loc= 2.35 spac=1.2
8   line x loc= 4.25 spac=0.25
9   line x loc= 4.50 spac=0.1
10  line x loc= 4.70 spac=0.1
11  line x loc= 4.95 spac=0.25
12  line x loc= 6.85 spac=1.2
13  line x loc= 8.75 spac=0.25
14  line x loc= 9.00 spac=0.25
15  #
16  line y loc=0.00 spac=0.04
17  line y loc=0.85 spac=0.125
18  line y loc=2.0   spac=0.5
19  line y loc=5.0   spac=1.2
20  #
21  # Define material, orientation, initial doping
22  init silicon boron conc=5e14 orientation=100
23  #
24  # Deposit oxide
25  #
26  deposit oxide thick=0.05 divisions=3
27  #
28  # deposit nitride
29  deposit nitride thick=0.05 divisions=3
30  #
31  # Implant phosphorus - n-well formation
32  implant phosphorus dose=1.4e12 energy=180
33  #
34  # Perform diffusion
35  #
36  method fermi compress
37  diffus time=320 temp=1000 nitro press=1.00
38  #
39  deposit poly thick=0.1
40  #
41  etch poly start x=4.5 y=-0.5
42  etch cont x=4.5 y=0.5
```

```
etch cont x=4.7 y=0.5
etch done x=4.7 y=-0.5
#
electrode name=gate1  x=0.0
electrode name=gate2  x=8.0
electrode name=substrate backside
#
extract name="junc_depth" xj silicon junc.occno=1 x.val=8

relax silicon y.min=2.0 dir.y=false

structure outf=ccdex02_0.str
tonyplot  ccdex02_0.str -set ccdex02_0.set
#
# Transfer the structure and start device simulation
#
go atlas
#
#
# SILVACO International 1994
#
# SECTION 1: Mesh and structure definition
#
# Read the structure generated in athena/ssuprem4
#
# (If run through the deckbuild beginning from process simulation
#  the structure is transfered automatically into device simulator:
#  no mesh statement is needed - commented out)
#
# mesh   inf=ccdex02_0.str master.in
#
# SECTION 2: Contacts
#
contact name=gate1 n.polysilicon
contact name=gate2 n.polysilicon
contact name=substrate
#
# SECTION 3: Material & Models
#
models   conmob fldmob srh print
#
# SECTION 4: Initial solution
#
```

```
method newton itlimit=60 trap carriers=0
solve init
#
# Stepping the gate1 simultaneously down to -8V:
#
solve vstep=-2 name=gate1 vfinal=-8
save outf=ccdx_gate1.str
#
# Stepping gate2 up to 12V
solve vstep=2 name=gate2 vfinal=12
save outf=ccdx_gate2.str
#
# Set up depleted well
#
solve n.bias=1
solve n.bias=2
solve n.bias=3
solve n.bias=4
solve n.bias=5
solve n.bias=6
solve n.bias=7
solve n.bias=8
solve n.bias=9
solve n.bias=10
solve n.bias=12
solve n.bias=14
solve n.bias=16
solve n.bias=18
solve n.bias=20
#
# Use LUMINOUS for carrier generation under light exposure
#
beam num=1 x=6.85 y=-1.0 min.w=-2.0 max.w=2.0 wavelength=0.8 angle=90.0
material material=Silicon real.index=1.45 imag.index=0.005

models   conmob fldmob srh print
method newton autonr trap itlimit=30 carriers=2

# short transient to see empty well
solve b1=0 tstep=1e-12 tstop=1e-12
#save empty structure

save outf=ccdex02_1.str
```

```
tonyplot  ccdex02_1.str -set ccdex02_1.set

# extract electron concentration in empty well (1.0e4 is to scale into I/
  cm^3)
extract init inf="ccdex02_1.str"
extract name="initial_n_at_x8" 1.0e+4*area from curve(depth,n.conc mate-
  rial="Silicon" mat.occno=1 x.val=8.0)


# shine light
solve b1=1.0 ramp.lit tstep=1e-9 tstop=1e-8 ramptime=1.e-8
save outf=ccd21e-8.str
solve b1=1.0 tstep=1e-8 tstop=1e-7
save outf=ccd21e-7.str
solve b1=1.0 tstep=1e-7 tstop=1e-6

save outf=ccdex02_2.str
tonyplot ccdex02_2.str -set ccdex02_2.set



# extract electron concentration in storage well (1.0e4 is to scale into
  I/cm^3)
extract init inf="ccdex02_2.str"
extract name="final_n_at_x8" 1.0e+4*area from curve(depth,n.conc materi-
  al="Silicon" mat.occno=1 x.val=8.0)
extract name="stored_n" $final_n_at_x8 - $initial_n_at_x8

quit
```

17.1.3 ccdex03.in: Finding the Maximum Potential and Integrated Charge

This example demonstrates postprocessing on CCD structures from the previous examples. It shows:

- extraction of the line of maximum potential from a 2D structure
- 2D integration of a charge packet

The first section of the example takes a solution file from the transient charge transfer in the first example in this section. It uses the extract feature `max.conc.file` to store a file of the maximum potential for each x location in the structure. This file can be plotted directly in TONYPLOT. Alternatively the file can be overlayed onto the 2D structure in TONYPLOT. This can be done by plotting the 2D CCD structure with the `flag-ccd`. Then the DISPLAY menu of TONYPLOT has an option to overlay the file saved by extract.

The second and third extract statement use the 2D integration feature of `extract` to give the integrated electron concentration in the charge packet. This is done for both empty and full well conditions

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

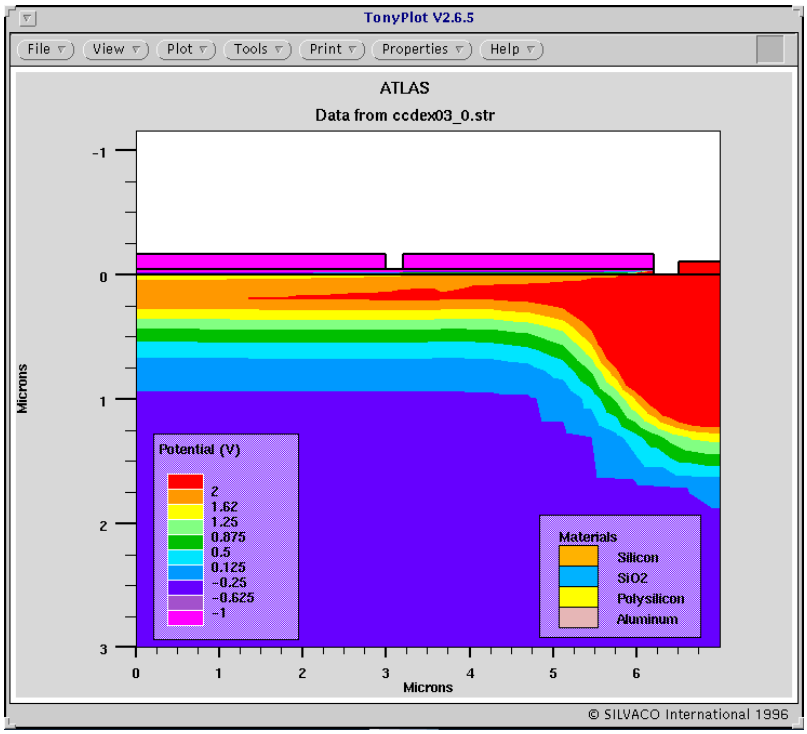


Figure 17.9: Potential in the structure from the first CCD example in the empty well state

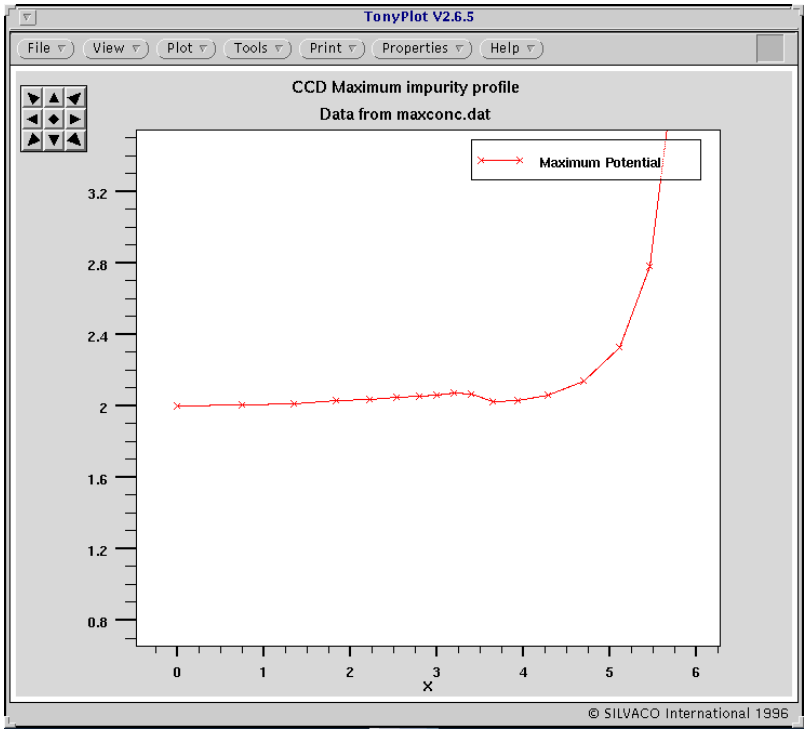


Figure 17.10: Peak potential at each vertical location in the device is plotted across the channel.

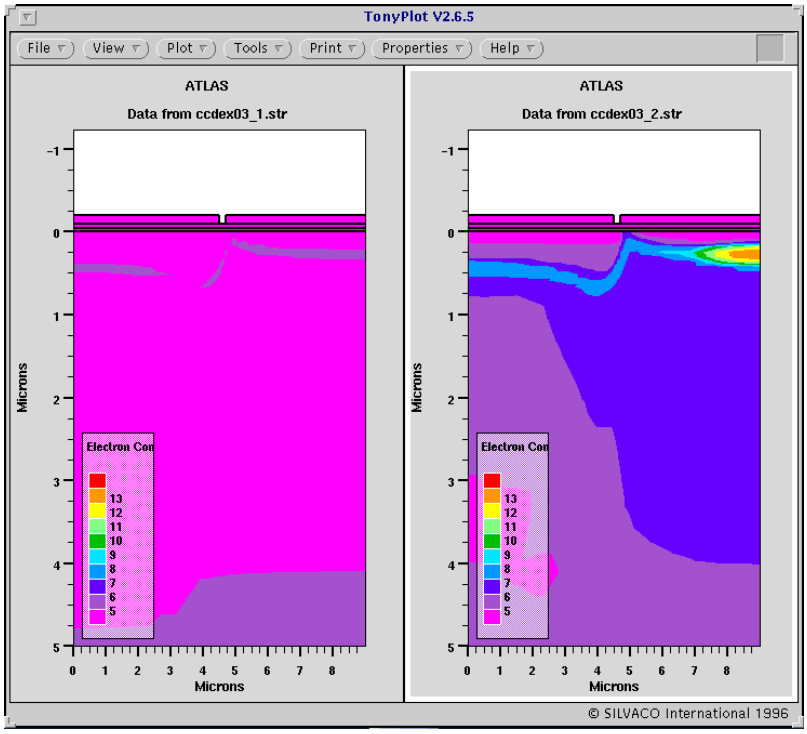


Figure 17.11: Comparison of empty and full well electron concentration for post-processing of net electrons stored

Input File ccd/ccdex03.in:

```
1   go atlas
2
3   # extract line of maximum potential from solution file
4
5   extract init inf="ccdex03_0.str"
6   extract name="maxpot" max.conc.file material="Silicon" impurity="Poten-
      tial" \
7   x.step=0.001 outfile="maxconc.dat" material="Silicon"
8   tonyplot -ccd -st ccdex03_0.str -set ccdex03_0.set
9   tonyplot maxconc.dat -set ccdex03_2.set
10
11
12  # 2D integration of electron concentration in empty well
13
14  extract init inf="ccdex03_1.str"
15  extract name="empty" 2d.area impurity="Electron Conc" x.step=0.25  \
16                       x.min=0.0 y.min=0.0 x.max=3 y.max=0.5
17
18
19  # 2D integration of electron concentration in full well
20
21  extract init inf="ccdex03_2.str"
22  extract name="full" 2d.area impurity="Electron Conc" x.step=0.25  \
23                       x.min=0.0 y.min=0.0 x.max=3 y.max=0.5
24
25  tonyplot -st  ccdex03_1.str ccdex03_2.str -set ccdex03_1.set
26
```

This page is intentionally left blank

18.1. LASER: Laser Diode Application Examples

18.1.1 laserex01.in : InP/InGaAsP Laser: Simple Gain Model

Requires: BLAZE/LASER

This example demonstrates simulation of a buried heterostructure InP/InGaAsP laser diode using the simple gain model. It shows:

- laser structure definition using ATLAS syntax
- material parameters specification
- setting conventional physical models
- device initial biasing
- activating LASER module of ATLAS
- mesh definition for solving the Helmholtz wave equation
- definition of laser physical models and their parameters including optical gain model
- calculation of laser electrical and optical characteristics

At the beginning of the input file the laser diode is described using ATLAS structural syntax including mesh, regions, electrodes and doping distributions the same way as it is done for any conventional applications. The `mesh` statement uses the `diag.flip` parameter to make the mesh symmetrical with respect to the vertical center cross section. The mesh symmetry reflects the respective symmetry of the structure and may be important for calculating correctly the light intensity, local optical gain, and other internal physical distributions. Mesh spacings are specified to resolve heterojunctions in x and y directions.

The laser structure is composed of 10 regions. The active lasing region of the device is made of quaternary InGaAsP compound and surrounded by InP regions of n- and p- type. All heterojunctions are modeled as abrupt. The cathode electrode is defined in the center of the top of the device. The anode electrode extends across the device at the bottom. The composition parameters of the active region x.comp and y.comp are defined in the respective `region` statement. The composition parameters are used to calculate fundamental properties of the InGaAsP material. e.g. energy band gap, dielectric constant, densities of states in the conduction and valence bands, etc. using the default dependencies. The default values of the above parameters are used also for the InP regions.

Some other material parameters are specified in the `material` statements separate for InP and InGaAsP materials. These include electron and hole lifetimes (capture times) for Shockley-Read-Hall recombination model, optical recombination constants, and low field mobilities. In addition, the electron affinity for InGaAsP is explicitly defined to ensure correct energy band alignment. The parameter, `gain1min`, describing local optical gain and different from the default value is also given in the `material` statement for InGaAsP.

In two `models` statements a standard set of the ATLAS/BLAZE models is specified. In order to save computational time, no LASER specific models are defined at this point. The LASER models are specified later in the input file after the initial bias has been increased. According to theory no laser emission will occur until the difference of quasi-Fermi levels in the active layer exceeds the material band gap. Therefore, it is recommended to ramp the bias up to some value not greater than the band gap of the active region, and then enable the LASER models. In this example after obtaining the initial solution the anode voltage is ramped up to 0.6 V in the two-carrier solution mode.

The next part of the input file enables LASER specific models. It starts from defining a mesh for solution of the Helmholtz wave equation. The Helmholtz equation is solved on a uniform rectangular mesh that is independent of the triangular mesh used for electrical equations. The rectangular

mesh and the region in which the Helmholtz equation is solved is defined using the `lx.mesh` and `ly.mesh` statements. The defined region must lie completely inside the BLAZE simulation domain and cover completely the active lasing region. The conventional models and LASER specific models are specified in the `models` statement. The LASER simulation is enabled by the parameter `laser` in this statement. In this example the the simple empirical gain model is activated by the parameter `gainmod=2`. Here the optical gain is a function of electron and hole concentrations given in the form of analytical expression with empirical coefficients. The parameter `las_maxch` specifies the maximum allowable relative change in the photon densities between iterations. If not limited the rapid change in the photon density may cause convergence problems. The parameter `photon_energy` is used to specify the energy of the photons. If `gainmod=2` is used, it specifies only an initial estimate of the photon energy. Instead of `photon_energy`, the parameter `las_omega` can be used to define the lasing frequency. The parameter, `cavity_length` specifies the cavity length in the 3rd z-direction. For other LASER models and their parameters the default dependencies and values are applied.

In the LASER mode the anode voltage is ramped up to 1.7V. The I-V, light power, and total optical gain data is saved in the log file. The final physical distributions in the structure are saved in the output structure file. The following results are visualized in TONYPLOT:

- conventional I-V characteristic
- light power versus current
- optical gain versus current
- light intensity distribution over the structure

The behavior of the optical gain for the fundamental transverse mode illustrates its increase until the current reaches the threshold value and then it remains flat equal to the light losses (in this case 30cm-1).

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

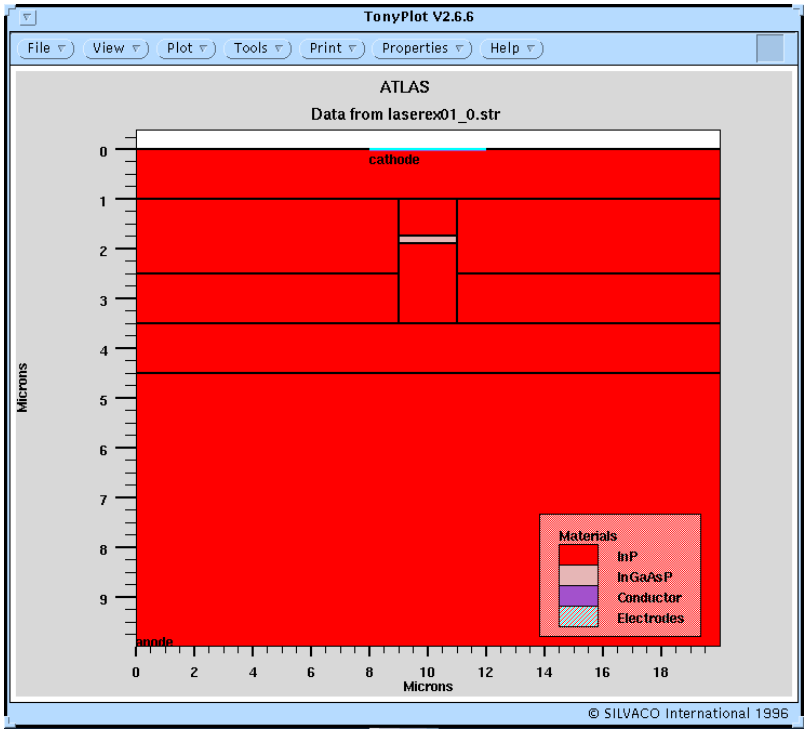


Figure 18.1: Structure and region boundaries of a InGaAsP LASER defined using ATLAS syntax. Light travels in the direction into/out of the page

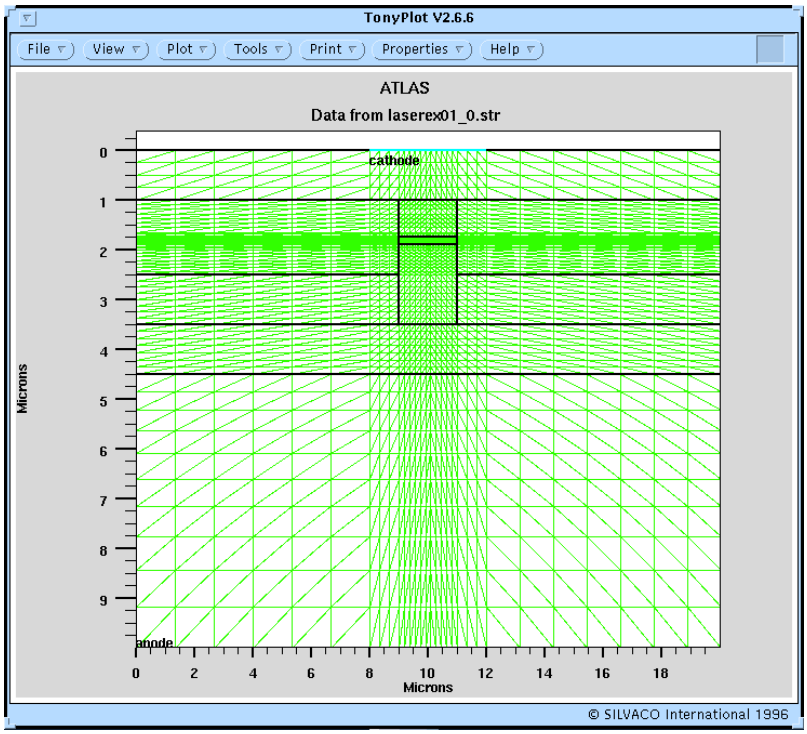


Figure 18.2: Mesh for LASER simulation. Note the mesh concentration in the active InGaAsP region

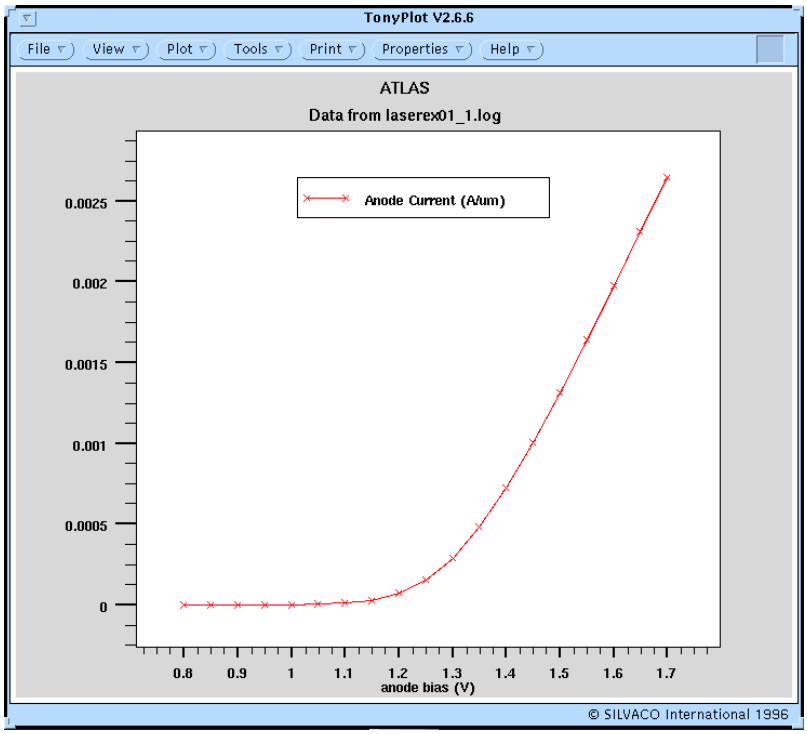


Figure 18.3: Anode Current vs bias in the LASER diode

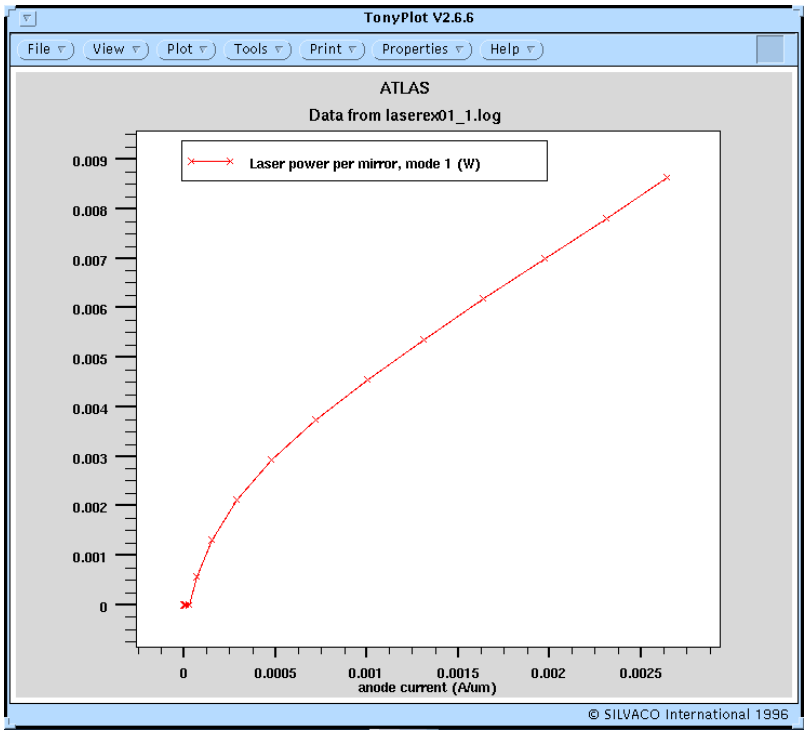


Figure 18.4: Laser power versus diode current

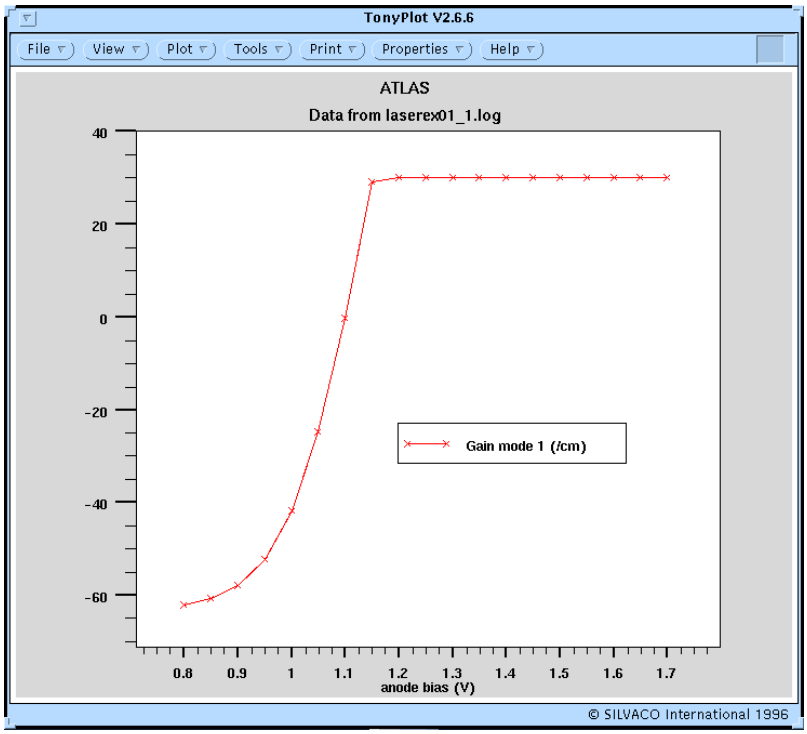


Figure 18.5: Optical Gain versus diode bias. Lasing action occurs once the gain is greater than unity

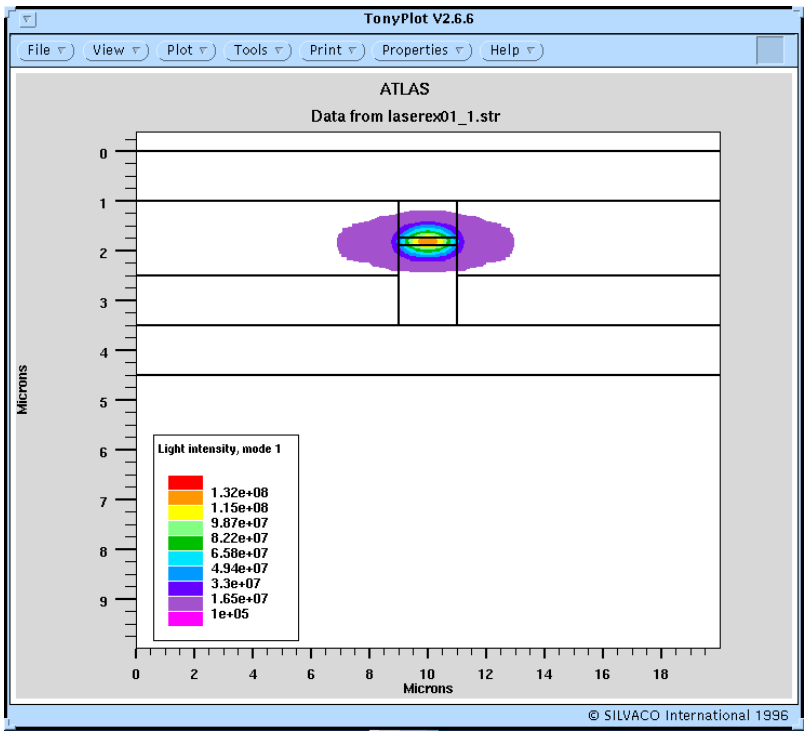


Figure 18.6: 2D display of near field light intensity over the structure.

Input File laser/laserex01.in:

```
1    go atlas
```

```
#
#      SILVACO International, 1993
#
#
mesh       rect  nx=31 ny=60  smooth=1 diag.flip
#
x.m        n=1     l=0.0          r=1.0
x.m        n=7     l=8.0          r=1.0
x.m        n=10    l=9.0          r=1.0
x.m        n=21    l=11.0         r=1.0
x.m        n=24    l=12.0         r=1.0
x.m        n=31    l=20.0         r=1.0
#
y.m        n=1     l=0.0          r=1.0
y.m        n=5     l=1.0          r=1.0
y.m        n=12    l=1.65         r=1.0
y.m        n=15    l=1.75         r=1.0
y.m        n=25    l=1.90         r=1.0
y.m        n=28    l=2.0          r=1.0
y.m        n=35    l=2.5          r=1.0
y.m        n=43    l=3.5          r=1.0
y.m        n=50    l=4.5          r=1.0
y.m        n=60    l=10.0         r=1.1
#
region     num=1  material=InP x.min=0. x.max=20.0 \
 y.min=0.0 y.max=1.0
#
region     num=2  material=InP x.min=0. x.max=9.0 \
 y.min=1.0 y.max=2.5
#
region     num=3  material=InP x.min=11.0 x.max=20.0 \
 y.min=1.0 y.max=2.5
#
region     num=4  material=InP x.min=9.0 x.max=11.0 \
 y.min=1.0 y.max=1.75
#
region     num=5  material=InGaAsP x.min=9.0 x.max=11.0 y.min=1.75 \
 y.max=1.9 x.comp=0.25 y.comp=0.5
#
region     num=6  material=InP x.min=0.0 x.max=9.0 \
 y.min=2.5 y.max=3.5
#
region     num=7  material=InP x.min=11.0 x.max=20.0 \
```

```
 y.min=2.5 y.max=3.5
#
region     num=8  material=InP x.min=9.0 x.max=11.0 \
 y.min=1.9 y.max=3.5
#
region     num=9  material=InP x.min=0.0 x.max=20.0 \
 y.min=3.5 y.max=4.5
#
region     num=10  material=InP x.min=0.0 x.max=20.0 \
 y.min=4.5 y.max=10.0
#
elec       num=1  name=cathode     \
  x.min=8.0 x.max=12.0 y.min=0.0 y.max=0.0
#
elec       num=2  name=anode       bottom
#
doping     uniform reg=1  n.type conc=1.e18
doping     uniform reg=2  p.type conc=2.e17
doping     uniform reg=3  p.type conc=2.e17
doping     uniform reg=4  n.type conc=1.e18
doping     uniform reg=5  p.type conc=2.e15
doping     uniform reg=6  n.type conc=2.e17
doping     uniform reg=7  n.type conc=2.e17
doping     uniform reg=8  p.type conc=1.e18
doping     uniform reg=9  p.type conc=1.e18
doping     uniform reg=10 p.type conc=2.e18


#
material material=InP     taun0=2.e-9  taup0=2.e-9  copt=1.5e-10 \
 mun=2400.0 mup=80.0
#
material material=InGaAsP taun0=10.e-9 taup0=10.e-9 copt=1.5e-10 \
 mun=4600.0 mup=150.0 affinity=3.12 gain1min=2.5e-16
#
models
models material=InP     fldmob  srh optr fermi print
models material=InGaAsP fldmob  srh optr fermi print
#
solve init

save outf=laserex01_0.str
tonyplot laserex01_0.str -set laserex01_0_str.set
```

```
88
89
90  #
91
92  method  newton autonr trap
93  solve v2=0.01
94  solve v2=0.05
95  solve v2=0.1
96  solve v2=0.2
97  solve v2=0.4
98  solve v2=0.6
99  #
100 #    LASER models
101 #
102 lx.m n=1  x=6.0
103 lx.m n=37 x=14.0
104 #
105 ly.m n=1  y=1.25
106 ly.m n=33 y=2.4
107 #
108 models material=InGaAsP fldmob  srh optr fermi print \
109  laser gainmod=2 las_maxch=5. \
110  photon_energy=1.025 \
111  cavity_length=50
112 #
113
114 log  outf=laserex01_1.log
115 #
116 solve v2=0.8
117 solve v2=0.9
118 solve v2=1.0
119 solve v2=1.1
120 #
121 output con.band val.band recomb u.srh u.aug u.rad flowlines
122 solve vstep=0.05 electr=2 vfinal=1.7
123 #save outfile=ind_ph_1.str
124 save outfile=laserex01_1.str
125 #
126
127 tonyplot  laserex01_1.log -set laserex01_1_log.set
128 tonyplot  laserex01_1.log -set laserex01_2_log.set
129 tonyplot  laserex01_1.log -set laserex01_3_log.set
130 tonyplot  laserex01_1.str -set laserex01_1_str.set
```

```
131
132 quit
133
134
135
136
```

18.1.2 laserex02.in: InP/InGaAsP Laser: Emission Spectrum Analysis

Requires: BLAZE/LASER

This example demonstrates simulation of a buried heterostructure InP/InGaAsP laser diode. Physically based frequency dependent optical gain model is used in this example to calculate light emission spectrum. The example shows:

- laser structure definition using ATLAS syntax
- setting up the device models and initial biasing
- activating LASER module of ATLAS
- mesh definition for solving the Helmholtz wave equation
- definition of laser physical models and their parameters including **frequency dependent optical gain model**
- calculation of laser electrical and optical characteristics

This example illustrates how ATLAS can be used to simulate spectral characteristics. The spectral characteristics refer to the longitudinal laser spectrum since LASER takes into account only the fundamental transverse mode solution. It is known from the theory that the longitudinal laser spectrum depends on the optical gain, spontaneous emission spectrum, and the cavity length in the third dimension (or Z-direction). A shorter cavity length leads to a wider energy spacing between adjacent longitudinal modes, so fewer longitudinal modes will be actually lasing. From a computational point of view, the situation of the nearly single mode laser operation is very desirable because each lasing mode considered imposes an additional computational burden.

For illustration purposes, the structure under consideration is the same as in the previous example in this section. We refer to that example regarding the laser structure definition, mesh, material and physical models specification.

The difference between this and the previous simulation is that physically based optical gain modes is activated in LASER simulation using the syntax `models lmodes gainmod=2. Gainmod=2` sets the frequency dependent model. The parameters, `las_einit, las_efinal` are used to specify the photon energy range for spectral analysis, and the `cavity_length` parameter is set to 50 microns. The later value is rather small but it allows us to demonstrate the spectral analysis within a comparatively short run time.

As in the previous example, the additional variables specified in the `output` statement will be saved in the output structure file. These include the conduction and valence band potentials for analyzing the energy band diagram, total, SRH, Auger, and radiative recombination rates and current flowlines.

During simulation, the IV data, light power and total optical gain data is stored in the log file. The laser spectra (gain spectra and photon density spectra) are stored in separate log files for each bias condition. The name of the spectrum log file is specified using the parameter `spec.name` in the `models` statement in the LASER portion of the input file. The following results are displayed in TONYPLOT:

- total light output power versus current

- gain spectra for different laser biases (currents) below and above lasing threshold
- light output spectrum below and above lasing threshold

For comparison the light output power vs diode current is shown along with the similar data obtained in the previous example. The gain saturation effect is clearly seen after laser threshold. Only a few of the longitudinal modes are actually lasing and the light output spectrum is practically the same after the lasing threshold.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

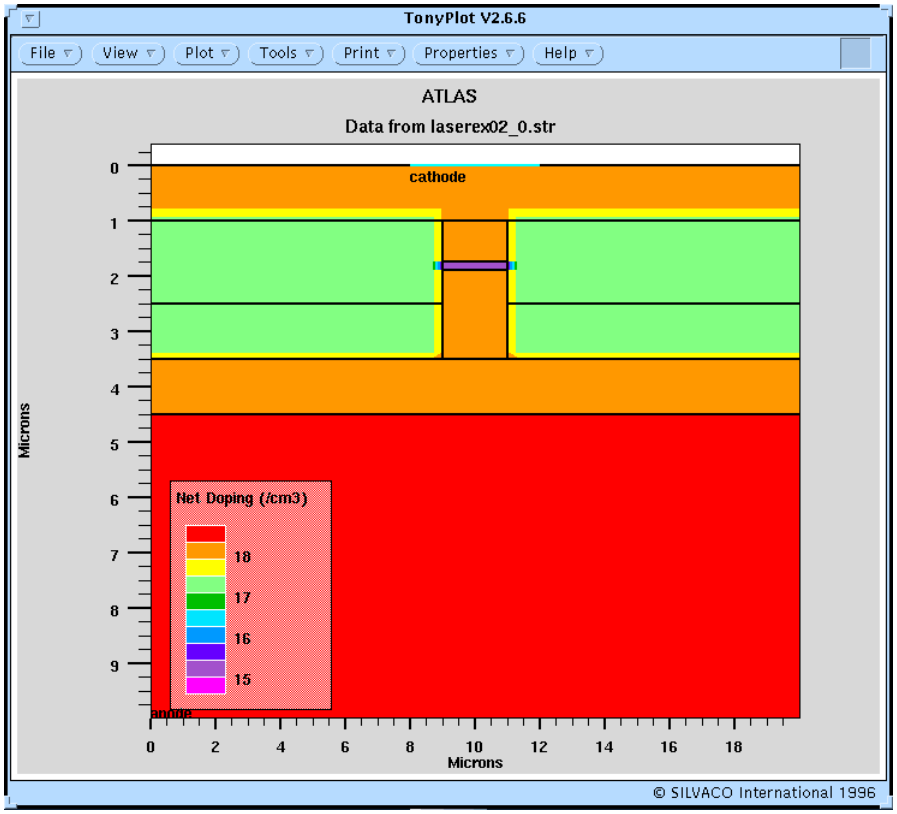


Figure 18.7: Doping concentration in the InGaAsP Laser Diode

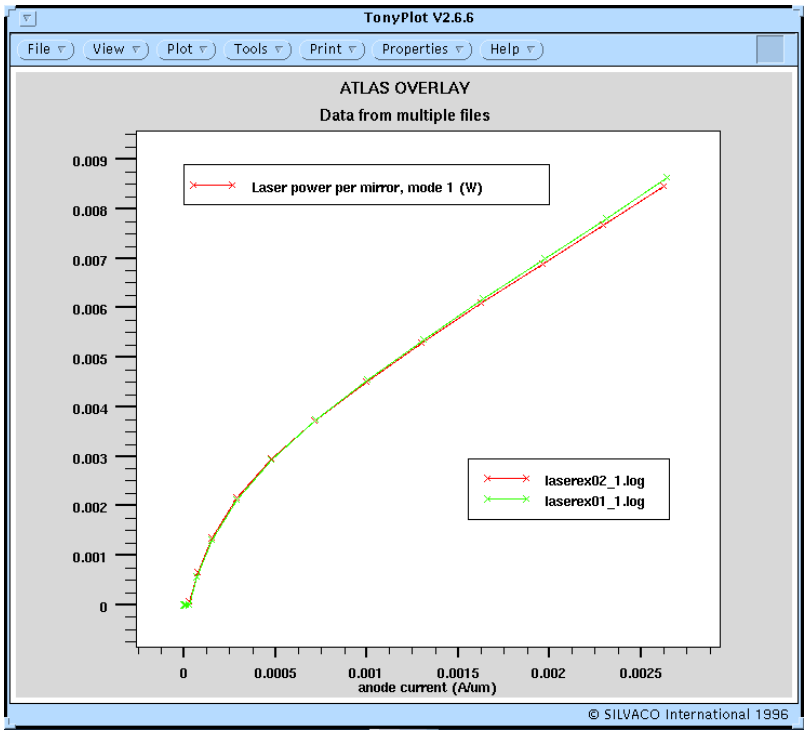


Figure 18.8: Comparison of the frequency dependent model versus the simpler model used in the previous example. Results are almost identical.

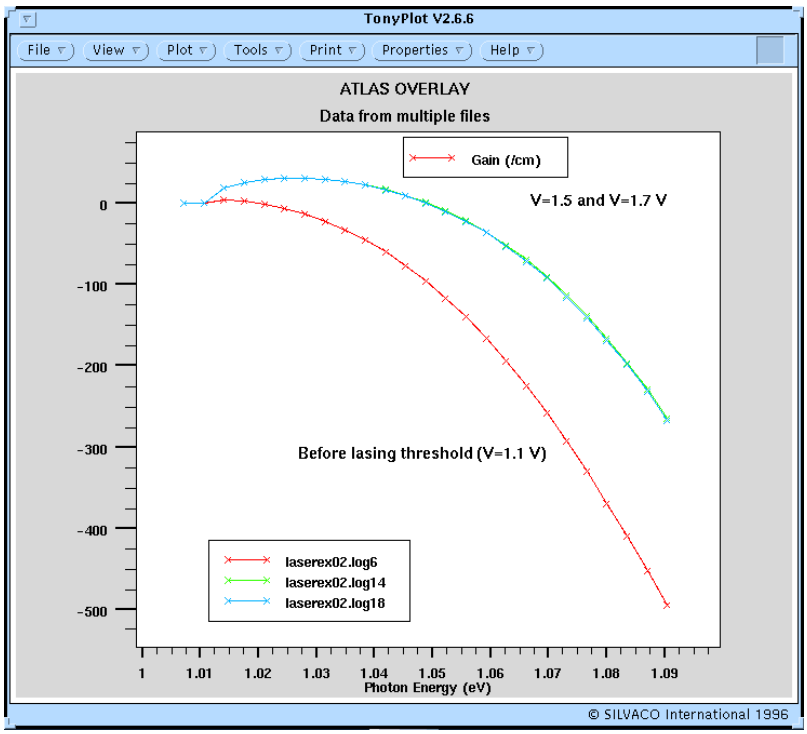


Figure 18.9: Gain versus photon energy for two different diode biases. Any Gain above unity will give lasing action .

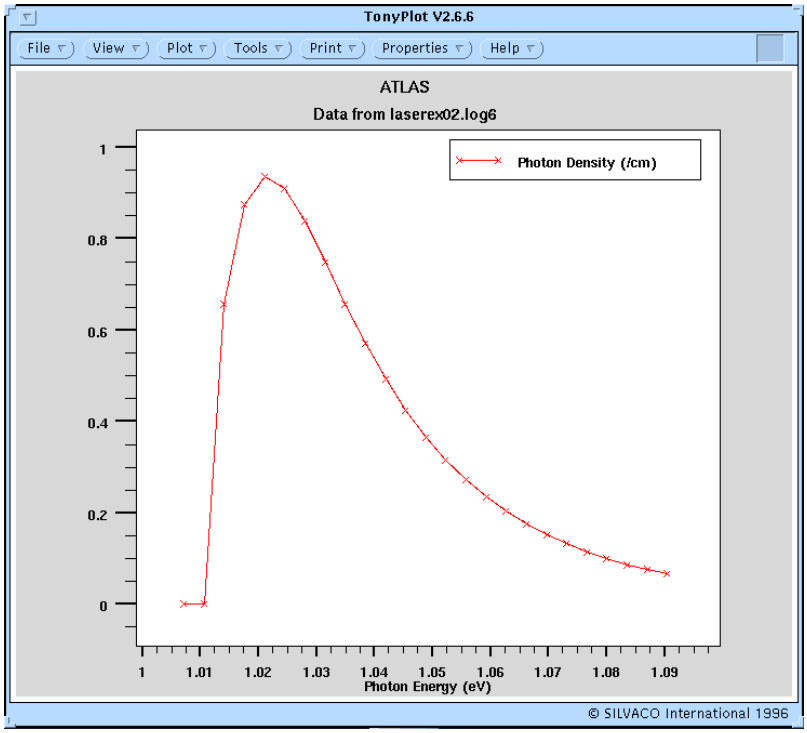


Figure 18.10: Spectral response of the LASER at low voltages where gain is < 1

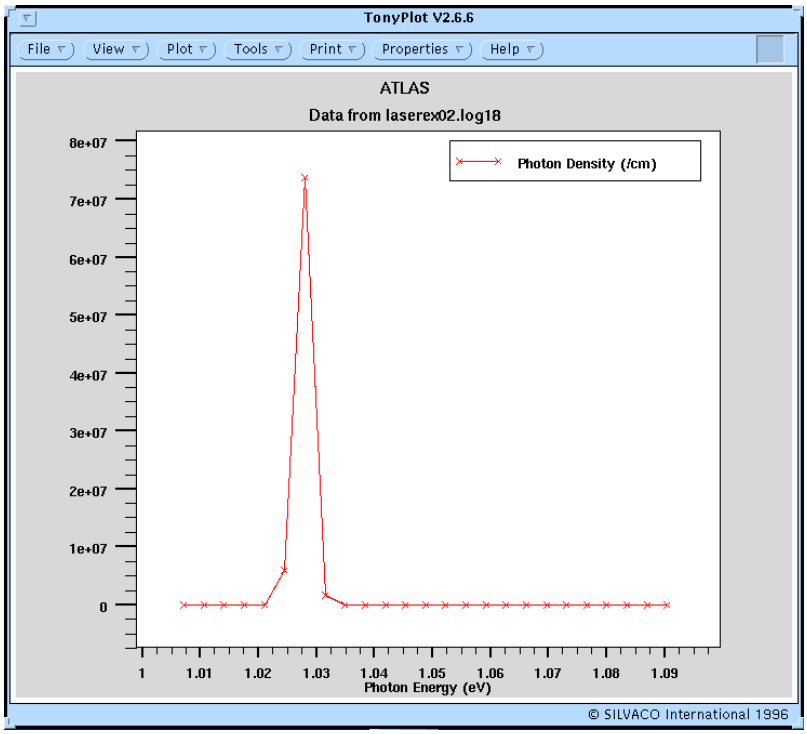


Figure 18.11: Spectral response of the LASER at higher voltages where gain is > 1

Input File laser/laserex02.in:

```
1    go atlas
```

```
#
#      SILVACO International, 1993
#
#
mesh       rect  nx=31 ny=60  smooth=1 diag.flip
#
x.m        n=1     l=0.0          r=1.0
x.m        n=7     l=8.0          r=1.0
x.m        n=10    l=9.0          r=1.0
x.m        n=21    l=11.0         r=1.0
x.m        n=24    l=12.0         r=1.0
x.m        n=31    l=20.0         r=1.0
#
y.m        n=1     l=0.0          r=1.0
y.m        n=5     l=1.0          r=1.0
y.m        n=12    l=1.65         r=1.0
y.m        n=15    l=1.75         r=1.0
y.m        n=25    l=1.90         r=1.0
y.m        n=28    l=2.0          r=1.0
y.m        n=35    l=2.5          r=1.0
y.m        n=43    l=3.5          r=1.0
y.m        n=50    l=4.5          r=1.0
y.m        n=60    l=10.0         r=1.1
#
region     num=1  material=InP x.min=0. x.max=20.0 \
 y.min=0.0 y.max=1.0
#
region     num=2  material=InP x.min=0. x.max=9.0 \
 y.min=1.0 y.max=2.5
#
region     num=3  material=InP x.min=11.0 x.max=20.0 \
 y.min=1.0 y.max=2.5
#
region     num=4  material=InP x.min=9.0 x.max=11.0 \
 y.min=1.0 y.max=1.75
#
region     num=5  material=InGaAsP x.min=9.0 x.max=11.0 y.min=1.75 \
 y.max=1.9 x.comp=0.25 y.comp=0.5
#
region     num=6  material=InP x.min=0.0 x.max=9.0 \
 y.min=2.5 y.max=3.5
#
region     num=7  material=InP x.min=11.0 x.max=20.0 \
```

```
 y.min=2.5 y.max=3.5
#
region     num=8  material=InP x.min=9.0 x.max=11.0 \
 y.min=1.9 y.max=3.5
#
region     num=9  material=InP x.min=0.0 x.max=20.0 \
 y.min=3.5 y.max=4.5
#
region     num=10  material=InP x.min=0.0 x.max=20.0 \
 y.min=4.5 y.max=10.0
#
elec       num=1  name=cathode       \
 x.min=8.0 x.max=12.0 y.min=0.0 y.max=0.0
#
elec       num=2  name=anode       bottom
#
doping     uniform reg=1  n.type conc=1.e18
doping     uniform reg=2  p.type conc=2.e17
doping     uniform reg=3  p.type conc=2.e17
doping     uniform reg=4  n.type conc=1.e18
doping     uniform reg=5  p.type conc=2.e15
doping     uniform reg=6  n.type conc=2.e17
doping     uniform reg=7  n.type conc=2.e17
doping     uniform reg=8  p.type conc=1.e18
doping     uniform reg=9  p.type conc=1.e18
doping     uniform reg=10 p.type conc=2.e18
#
material material=InP     taun0=2.e-9  taup0=2.e-9  copt=1.5e-10 \
 mun=2400.0 mup=80.0
#
material material=InGaAsP taun0=10.e-9 taup0=10.e-9 copt=1.5e-10 \
 mun=4600.0 mup=150.0 affinity=3.12
#
models
models material=InP     fldmob  srh optr fermi print
models material=InGaAsP fldmob  srh optr fermi print
#

solve init
save outf=laserex02_0.str
tonyplot laserex02_0.str -set laserex02_0_str.set
#

```

```
88  method newton autonr trap
89  solve v2=0.01
90  solve v2=0.05
91  solve v2=0.1
92  solve v2=0.2
93  solve v2=0.4
94  solve v2=0.6
95  #
96  #    LASER models
97  #
98  lx.m n=1  x=6.0
99  lx.m n=37 x=14.0
100 #
101 ly.m n=1  y=1.25
102 ly.m n=33 y=2.4
103 #
104 models material=InGaAsP fldmob  srh optr fermi print \
105  laser gainmod=1 \
106  photon_energy=1.025 spec.name=laserex02.log \
107  lmodes las_einit=1.01 las_efinal=1.1 cavity_length=50
108 #
109
110 log  outf=laserex02_1.log
111 #
112 solve v2=0.8
113 solve v2=0.9
114 solve v2=1.0
115 solve v2=1.1
116 #
117 output con.band val.band recomb u.srh u.aug u.rad flowlines
118 solve vstep=0.05 electr=2 vfinal=1.7
119 save outfile=laserex02_1.str
120 #
121
122
123 tonyplot -overlay laserex02_1.log laserex01_1.log -set
      laserex02_1_log.set
124 tonyplot -overlay laserex02.log6 laserex02.log14 laserex02.log18 -set
      laserex02_2_log.set
125 tonyplot  laserex02.log6 -set laserex02_3_log.set
126 tonyplot  laserex02.log18 -set laserex02_4_log.set
127
128 quit
129
```

18.1.3 laserex03.in: GaAs/AlGaAs Strip Geometry Laser

Requires: BLAZE/LASER

This example demonstrates simulation of a the strip geometry GaAs/AlGaAs laser diode. Physically based frequency dependent optical gain model is used in this example. The example shows:

- laser structure definition using ATLAS syntax
- material parameters specification
- setting conventional physical models
- device initial biasing
- activating LASER module of ATLAS
- mesh definition for solving the Helmholtz wave equation
- definition of laser physical models and their parameters including optical gain model
- calculation of laser electrical and optical characteristics

In this example, the same approach is applied to simulate a laser structure of a different geometry and material composition. The active region of the device under consideration represents a narrow strip made of GaAs concluded between two AlGaAs regions. The logistics of the structure, material parameters and physical models definition repeat that of the previous two examples in this section. Refer to those examples for more detailed explanations.

After performing LASER simulation and saving the results in the log and structure output files the following results are displayed in TONYPLOT:

- total light output power versus current
- light intensity distribution over the structure

Since strip geometry lasers do not provide sufficient light confinement in the direction parallel to the p-n- and heterojunction planes, the lasing region can be smeared laterally along the active region or shifted to a side of the structure due to fluctuations in the carrier density and dielectric permittivity. The latter may lead to the spatial hole burning effect and have a negative influence on laser characteristics. In this particular example spatial hole burning does not occur, but the lack of index waveguiding in the lateral direction which results in spreading the lasing area along the active region is clearly visible.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

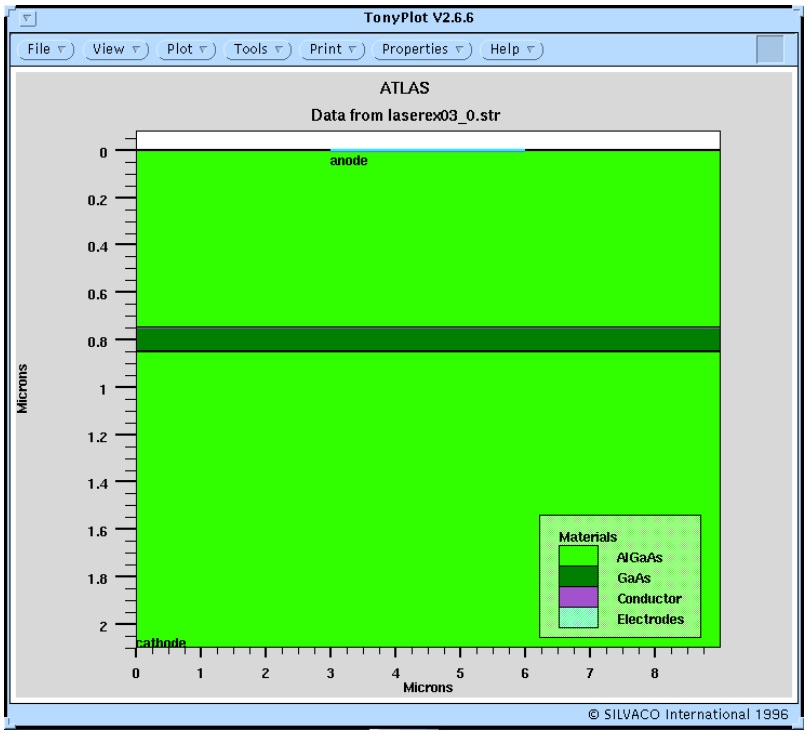


Figure 18.12: Geometry of an AlGaAs Strip LASER diode

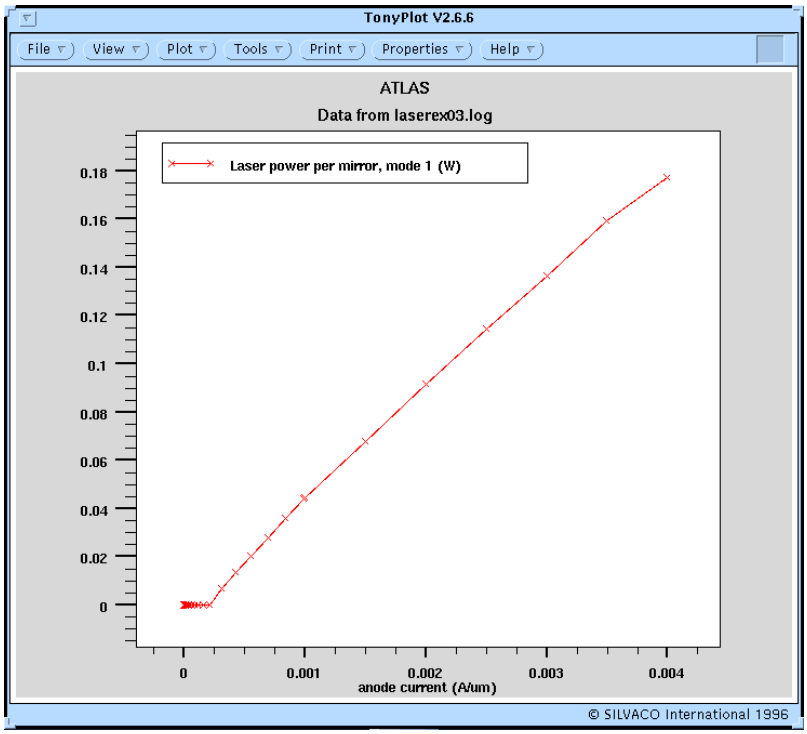


Figure 18.13: Diode IV curve for the strip LASER

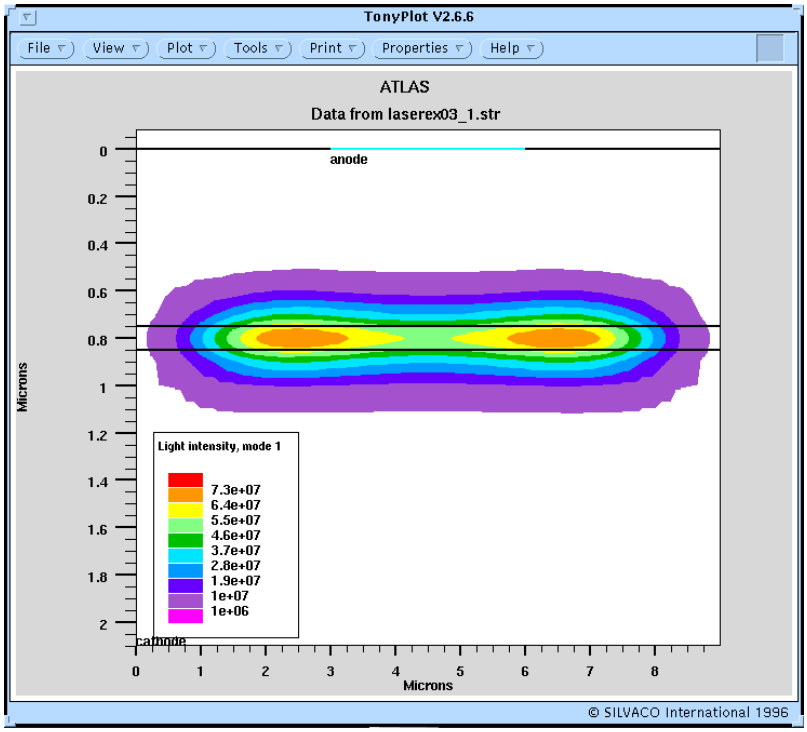


Figure 18.14: 2D optical intensity distribution for the strip laser. The shape is typical when the active region is not confined in the x direction

Input File laser/laserex03.in:

```
1   go atlas
2   #
3   #    SILVACO INTERNATIONAL, 1993
4   #
5   #    Illustration of lateral mode instability
6   #    in AlGaAs/GaAs strip geometry laser diode
7   #
8   mesh       rect  nx=31 ny=40  smooth=1 diag.flip
9   #
10  x.m        n=1     l=0.0          r=1.0
11  x.m        n=31    l=9.0          r=1.0
12  #
13  y.m        n=1     l=0.0          r=1.0
14  y.m        n=10    l=0.7          r=1.0
15  y.m        n=13    l=0.75         r=1.0
16  y.m        n=27    l=0.85         r=1.0
17  y.m        n=30    l=0.9          r=1.0
18  y.m        n=40    l=2.1          r=1.0
19  #
20  #
```

```
region     num=1  material=AlGaAs x.min=0. x.max=9.0 \
 y.min=0.0 y.max=0.75 x.composition=0.5
#
region     num=2  material=GaAs x.min=0.0 x.max=9.0 \
 y.min=0.75 y.max=0.85
#
region     num=3  material=AlGaAs x.min=0. x.max=9.0 \
 y.min=0.85 y.max=2.1 x.composition=0.5
#
elec       num=1  name=anode     x.min=3.0 x.max=6.0 y.min=0.0 y.max=0.0
elec       num=2  name=cathode   bottom
#
doping     uniform reg=1  p.type conc=1.e18
doping     uniform reg=2  n.type conc=4.e15
doping     uniform reg=3  n.type conc=4.e17
#
material material=GaAs   taun0=1.e-9 taup0=1.e-9 copt=1.5e-10
material material=AlGaAs taun0=1.e-9 taup0=1.e-9 copt=1.5e-10
#
model
model material=GaAs     fldmob srh optr print
model material=AlGaAs   fldmob srh optr
#
solve init
save outf=laserex03_0.str
tonyplot laserex03_0.str -set laserex03_0.set
#

method newton autonr trap  maxtrap=6 ir.tol=1.e-20 ix.tol=1.e-20

solve vanode=0.1
solve vanode=0.2
solve vanode=0.4 vstep=0.4 electr=1 vfinal=1.2
#
lx.m  n=1  x=0.1
lx.m  n=37 x=8.9
#
ly.m  n=1  y=0.5
ly.m  n=33 y=1.1
#
models material=GaAs fldmob  srh optr fermi print \
 laser gainmod=1 \
 las_xmin=0.1 las_xmax=8.9 las_ymin=0.5 las_ymax=1.1 \
```

```
 photon_energy=1.43 las_nx=37 las_ny=33 spec.name=laserex03sp.log \
 lmodes las_einit=1.415 las_efinal=1.47 cavity_length=100
#
log outf=laserex03.log
#
solve prev vstep=0.025 electr=1 vfinal=2.0
#
contact num=1 current
#
solve prev istep=0.0005 i1=0.001 ifinal=0.004 electr=1
save outfile=laserex03_1.str
#

tonyplot  laserex03.log -set laserex03_log.set


tonyplot laserex03_1.str -set laserex03_1.set

quit

```

19.1. SEU: Single Event Upset Application Examples

19.1.1 seuex01.in: SEU in a Reverse-Biased Diode

Requires: DEVEDIT3D, DEVICE3D

This examples demonstrates SEU simulation in a reversed biased three dimensional diode.

In this example a diode structure is constructed using DEVEDIT3D. The structure is then passed to ATLAS. The input file consists of the following parts:

- construction of the 3D device in DEVEDIT3D
- transfer of the structure to ATLAS by the DECKBUILD autointerface feature
- setting of the physical models
- perform dc bias solution
- specification of parameters for the charge track
- simulation of the effects of SEU

The first stage of the input constructs the diode geometry, material regions, doping profiles and electrodes in DEVEDIT3D. The structure was created in DEVEDIT3D by drawing the device regions in interactive mode and specifying 3D doping distribution. Finally the mesh was generated automatically by specifying basic mesh constraints and refining in the important areas of the device.

The ATLAS simulation begins by reading the structure from DEVEDIT3D. DECKBUILD provides autointerface between DevEdit3D and ATLAS so that the structure produced by the DEVEDIT3D is transferred to ATLAS without having to indicate the mesh statement.

The `models` statement is used to select a set of physical models for this simulation. In this case, these models are SRH recombination, bandgap narrowing, concentration and field dependent mobility model.

After the initial solution, the emitter voltage is ramped up to 3V in DC mode, thereby giving the initial condition of the structure for transient analysis. The Newton algorithm is used for these calculations.

The key syntax in any SEU simulation is the `singleeventupset` statement, which is used to specify entry and exit points, radius and density of the electron-hole pair distribution generated by the particle track.

Transient analysis is then performed to simulate charge collection in the device due to the SEU. The `method halfimpl` statement indicates that the half implicit scheme is to be used in the transient simulation. In this scheme the Poisson's two continuity and the total current equations are solved in sequence. This method ensures efficient and fast convergence in transient mode.

The results of the simulation are saved in the log file and then displayed using TonyPlot.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

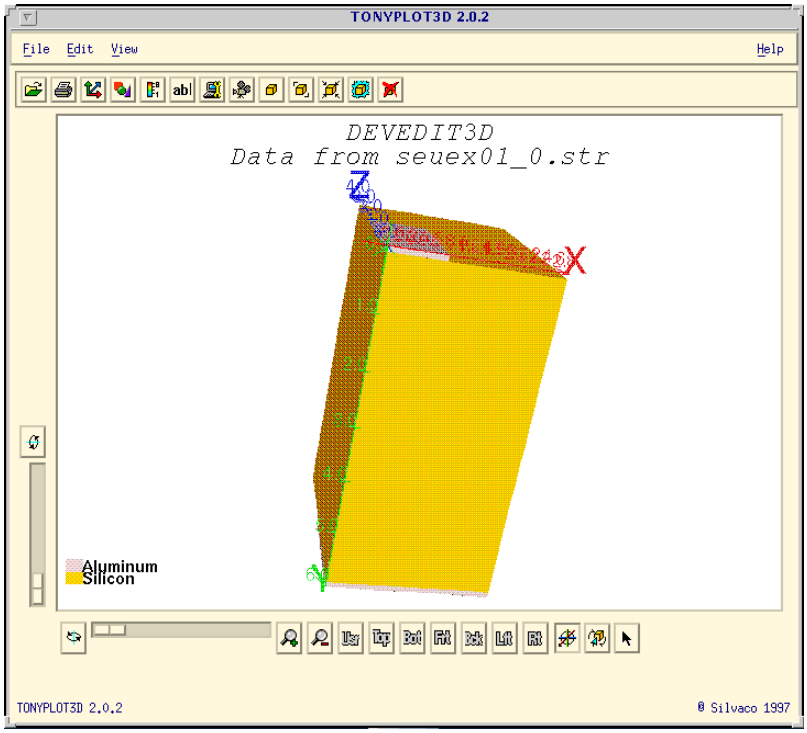


Figure 19.1: 3D diode structure defined in DevEdit3D

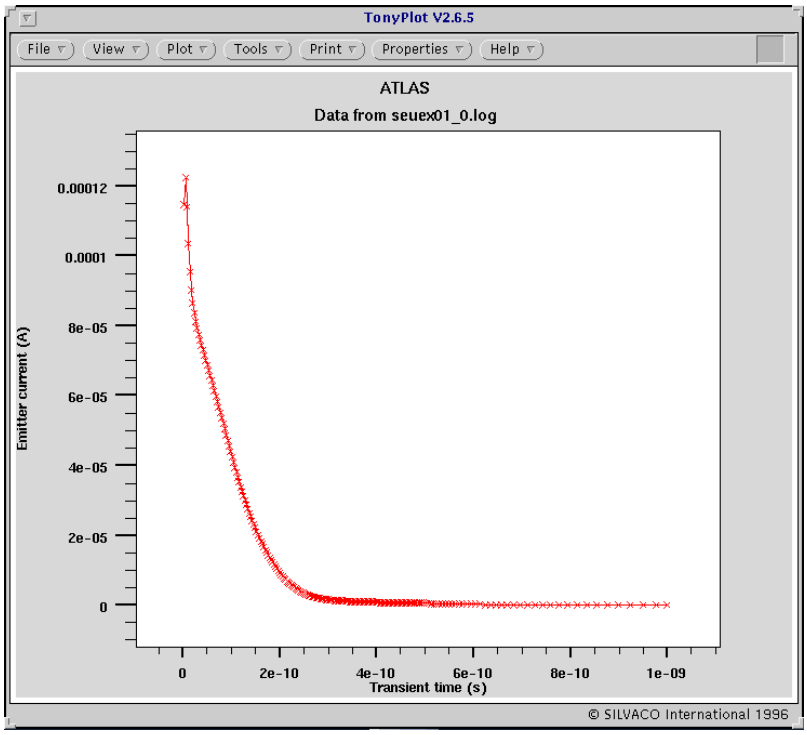


Figure 19.2: Diode current versus time after a single SEU strike

Input File seu/seuex01.in :

```
1    go DevEdit
```

```
2
3   #
4   #  pn diode seu simulation
5   #
6   DevEdit version="2.0" library="1.14"
7
8   work.area left=0 top=-0.1 right=3 bottom=6.1
9
10  # SILVACO Library V1.14
11
12  region reg=1 mat=Silicon color=0xffc000 pattern=0x3 Z1=0 Z2=4 \
13  points="0,0 1,0 3,0 3,6 0,6 0,0"
14  #
15  impurity id=1 region.id=1 imp=Boron color=0x906000 \
16  x1=0 x2=0 y1=0 y2=0 \
17  peak.value=2e+15 ref.value=0 z1=0 z2=0 comb.func=Multiply \
18  rolloff.y=both conc.func.y=Constant \
19  rolloff.x=both conc.func.x=Constant \
20  rolloff.z=both conc.func.z=Constant
21  #
22  constr.mesh region=1 default
23
24  region reg=2 name=Emitter mat=Aluminum elec.id=1 color=0xffc8c8 pat-
      tern=0x6 Z1=0 Z2=2 \
25  points="1,-0.1 1,0 0,0 0,-0.1 1,-0.1"
26  #
27  constr.mesh region=2 default
28
29  region reg=3 name=base mat=Aluminum elec.id=2 work.func=0 color=0xffc8c8
      pattern=0x6 Z1=0 Z2=2 \
30  points="0,6 3,6 3,6.1 0,6.1 0,6"
31  #
32  constr.mesh region=3 default
33
34
35  impurity id=1 imp=Arsenic color=0x906000 \
36  x1=0 x2=1.5 y1=0 y2=0 \
37  peak.value=1e+18 ref.value=0 z1=0 z2=2.5 comb.func=Multiply \
38  rolloff.y=both conc.func.y=Gaussian conc.param.y=0.2 \
39  rolloff.x=both conc.func.x="Error Function" conc.param.x=0.1 \
40  rolloff.z=both conc.func.z="Error Function" conc.param.z=0.1
41
42  # Set Meshing Parameters
43  #
```

```
44 base.mesh height=0.5 width=0.5
45 #
46 bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=0.001
     line.straightening=1 align.points when=automatic
47 #
48 imp.refine imp="Arsenic" sensitivity=0.8
49 imp.refine min.spacing=0.01 z=0
50 #
51 constr.mesh max.angle=90 max.ratio=300 max.height=1 \
52 max.width=1 min.height=0.0001 min.width=0.0001
53 #
54 constr.mesh type=Semiconductor default
55 #
56 constr.mesh type=Insulator default
57 #
58 constr.mesh type=Metal default
59 #
60 constr.mesh type=Other default
61 #
62 constr.mesh region=1 default
63 #
64 constr.mesh region=2 default
65 #
66 constr.mesh region=3 default
67 #
68 # Perform mesh operations
69 #
70 Mesh Mode=MeshBuild
71
72 imp.refine imp="Arsenic" sensitivity=0.8
73 imp.refine min.spacing=0.01 z=0
74
75 constr.mesh max.angle=90 max.ratio=300 max.height=1 \
76 max.width=1 min.height=0.0001 min.width=0.0001
77 #
78 constr.mesh type=Semiconductor default
79 #
80 constr.mesh type=Insulator default
81 #
82 constr.mesh type=Metal default
83 #
84 constr.mesh type=Other default
85
```

```
86  z.plane z=0 spacing=0.5
87  #
88  z.plane z=2 spacing=0.5
89  #
90  z.plane z=2.25 spacing=0.25
91  #
92  z.plane z=2.5 spacing=0.1
93  #
94  z.plane z=3 spacing=0.1
95  #
96  z.plane z=4 spacing=0.25
97  #
98  z.plane max.spacing=1000000 max.ratio=10
99
100 base.mesh height=0.5 width=0.5
101
102 bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=0.001
      line.straightening=1 align.Points when=automatic
103 structure outf=seuex01_0.str
104
105 go atlas
106
107 # set the models
108 models   bgn srh conmob fldmob
109 method newton
110
111 # set up initial DC bias
112 solve v1=0.5
113 solve v1=3.0
114
115 # set up the track of the radiation for SEU
116 singleeventupset entry="2.0,1.5,0." exit="1.,1.,4" radius=0.05 densi-
      ty=1e18
117
118 # use half implicit scheme
119 method halfimpl dt.min=3.e-12
120
121 # put the transient results in a file
122 log outf=seuex01_0.log
123
124 # perform the transient calculation
125 solve tfinal=1.e-9 tstep=3.e-12
126
127
```

```
128 tonyplot seuex01_0.log -set seuex01_0_log.set
129
130 quit
131
```

19.1.2 seuex02.in: Multiple Ion Tracks

Requires: DEVEDIT3D/DEVICE3D

This examples demonstrates SEU simulation in a diode structure in three dimensions initiated by multiple tracks. Time dependence of the carrier generation is also included in the specification of the second particle track. The input file consists of the following parts:

- construction of the 3D device in DEVEDIT3D
- transfer of the structure to ATLAS by DeckBuild autointerface feature
- physical models specification
- SEU parameter specification
- simulation of the effects of SEU

The diode structure under consideration is the same as in the previous example and is constructed with the same DevEdit3D run. The structure is then passed to ATLAS for electrical calculations using the autointerface feature of the DevEdit3D. The same set of physical models is applied.

The two `singleeventupset` statements are used now to specify entry and exit points, radius and density of generated electron-hole pairs due to two alpha particle tracks. Different from the first track, where the generation rate is position dependent only, the second track includes position and time dependent generation. Parameters `tc` and `t0` in the second statement specify the characteristic time of the Gaussian time dependency, and the time instant of the peak of generation.

The emitter voltage is ramped here again up to 3V in the DC mode, and then the transient analysis is performed to simulate charge collection in the device. The numerical algorithms are used to repeat those of the previous example.

The results of the simulation are saved in the log file and then displayed using TONYPLOT.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

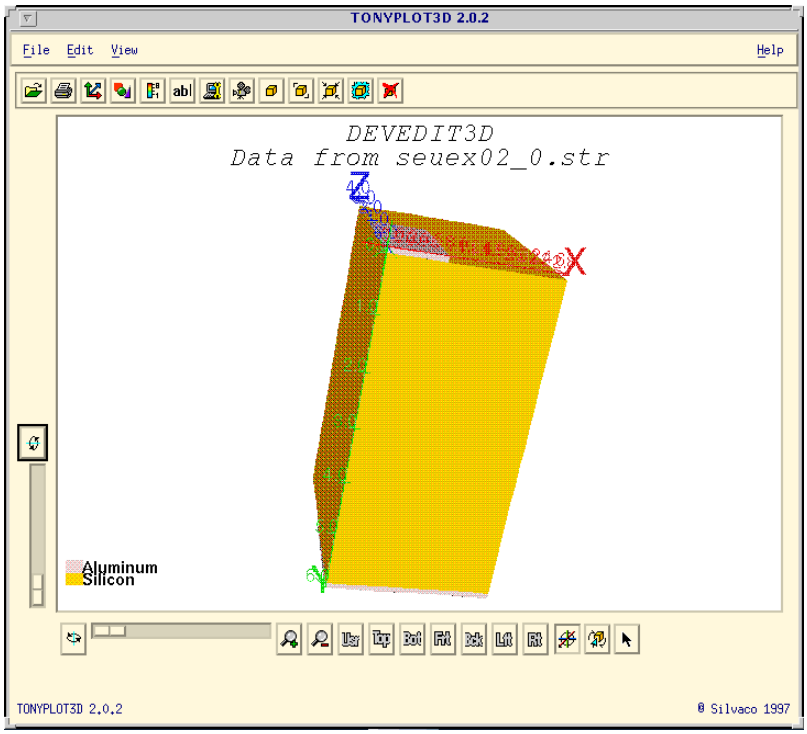


Figure 19.3: 3D diode structure defined in DevEdit3D

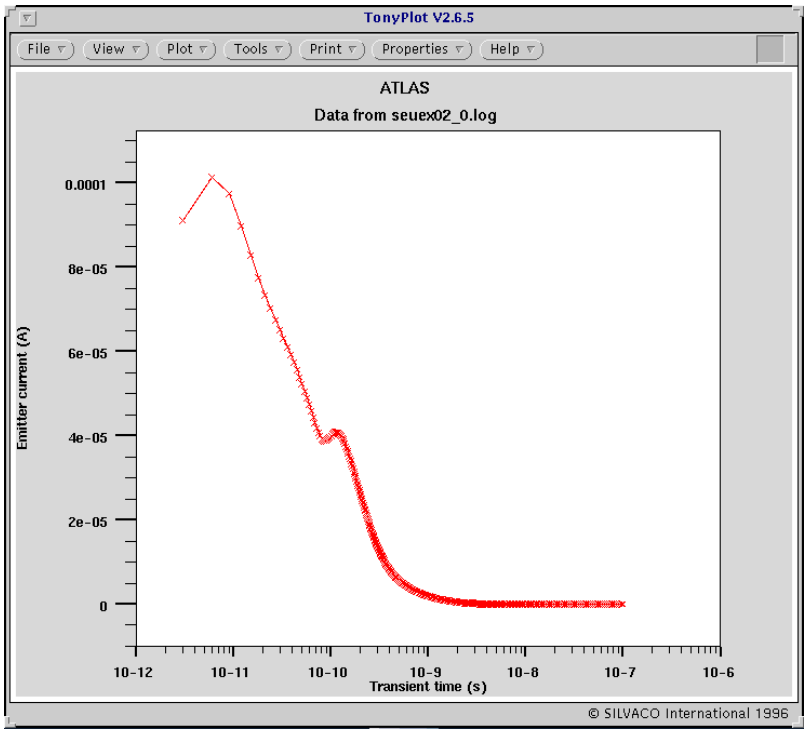


Figure 19.4: Diode current versus time after a multiple SEU strikes offset in time

Input File seu/seuex02.in:

```
1    go DevEdit
```

```
2
3   #
4   #  pn diode seu simulation with multiple tracks
5   #
6   DevEdit version="2.0" library="1.14"
7
8   work.area left=0 top=-0.1 right=3 bottom=6.1
9
10  # SILVACO Library V1.14
11
12  region reg=1 mat=Silicon color=0xffc000 pattern=0x3 Z1=0 Z2=4 \
13  points="0,0 1,0 3,0 3,6 0,6 0,0"
14  #
15  impurity id=1 region.id=1 imp=Boron color=0x906000 \
16  x1=0 x2=0 y1=0 y2=0 \
17  peak.value=2e+15 ref.value=0 z1=0 z2=0 comb.func=Multiply \
18  rolloff.y=both conc.func.y=Constant \
19  rolloff.x=both conc.func.x=Constant \
20  rolloff.z=both conc.func.z=Constant
21  #
22  constr.mesh region=1 default
23
24  region reg=2 name=Emitter mat=Aluminum elec.id=1 color=0xffc8c8 pat-
      tern=0x6 Z1=0 Z2=2 \
25  points="1,-0.1 1,0 0,0 0,-0.1 1,-0.1"
26  #
27  constr.mesh region=2 default
28
29  region reg=3 name=base mat=Aluminum elec.id=2 work.func=0 color=0xffc8c8
      pattern=0x6 Z1=0 Z2=2 \
30  points="0,6 3,6 3,6.1 0,6.1 0,6"
31  #
32  constr.mesh region=3 default
33
34
35  impurity id=1 imp=Arsenic color=0x906000 \
36  x1=0 x2=1.5 y1=0 y2=0 \
37  peak.value=1e+18 ref.value=0 z1=0 z2=2.5 comb.func=Multiply \
38  rolloff.y=both conc.func.y=Gaussian conc.param.y=0.2 \
39  rolloff.x=both conc.func.x="Error Function" conc.param.x=0.1 \
40  rolloff.z=both conc.func.z="Error Function" conc.param.z=0.1
41
42  # Set Meshing Parameters
43  #
```

```
44 base.mesh height=0.5 width=0.5
45 #
46 bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=0.001
     line.straightening=1 align.points when=automatic
47 #
48 imp.refine imp="Arsenic" sensitivity=0.8
49 imp.refine min.spacing=0.01 z=0
50 #
51 constr.mesh max.angle=90 max.ratio=300 max.height=1 \
52 max.width=1 min.height=0.0001 min.width=0.0001
53 #
54 constr.mesh type=Semiconductor default
55 #
56 constr.mesh type=Insulator default
57 #
58 constr.mesh type=Metal default
59 #
60 constr.mesh type=Other default
61 #
62 constr.mesh region=1 default
63 #
64 constr.mesh region=2 default
65 #
66 constr.mesh region=3 default
67 #
68 # Perform mesh operations
69 #
70 Mesh Mode=MeshBuild
71
72 imp.refine imp="Arsenic" sensitivity=0.8
73 imp.refine min.spacing=0.01 z=0
74
75 constr.mesh max.angle=90 max.ratio=300 max.height=1 \
76 max.width=1 min.height=0.0001 min.width=0.0001
77 #
78 constr.mesh type=Semiconductor default
79 #
80 constr.mesh type=Insulator default
81 #
82 constr.mesh type=Metal default
83 #
84 constr.mesh type=Other default
85
```

```
z.plane z=0 spacing=0.5
#
z.plane z=2 spacing=0.5
#
z.plane z=2.25 spacing=0.25
#
z.plane z=2.5 spacing=0.1
#
z.plane z=3 spacing=0.1
#
z.plane z=4 spacing=0.25
#
z.plane max.spacing=1000000 max.ratio=10

base.mesh height=0.5 width=0.5

bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=0.001
  line.straightening=1 align.Points when=automatic
structure outf=seuex02_0.str


go atlas

models   bgn srh conmob fldmob
#
singleeventupset entry="2.0,1.5,0." exit="1.,1.,4" density=7.e17 radi-
  us=0.05
singleeventupset entry="2.5,0.0,2.5" exit="1.5,8,3" radius=0.04 densi-
  ty=1.e18 \
t0=1.e-10 tc=2.e-11
#
method newton
#
solve init
#
solve v1=0.5
solve v1=3.0
#
method halfimpl dt.min=3.e-12
log outf=seuex02_0.log
solve tfinal=1.e-7 tstep=3.e-12
#
tonyplot seuex02_0.log -set seuex02_0_log.set
#
```

```
127 quit
128
129
```

19.1.3 seuex03.in: Single Event Upset in a 3D MOSFET

Requires: DEVEDIT3D/DEVICE3D

The current example demonstrates simulation of the effects of different angles of incidence of an SEU in a MOSFET structure. The MOSFET structure is constructed using DEVEDIT3D. The structure is then passed to ATLAS for electrical testing. The input file consists of the following parts:

- construction of the device in DEVEDIT3D
- SEU simulation in ATLAS: particle with normal incidence
- SEU simulation in ATLAS: particle with oblique incidence

ATLAS contains many features to enable the simulation of Single Event Upset (SEU) phenomena. Models for charge generation exist both in 2D, 3D and quasi-3D cylindrical coordinates. It is also possible for users to add their own position and time dependent charge generation using the C-Interpreter and to incorporate these into ATLAS simulations.

The first stage of the input file constructs the MOSFET geometry, material regions, doping profiles, electrodes and subsequently generates the mesh in 3D. This is done in DevEdit3D by drawing the device regions in interactive mode, and specifying 3D doping distributions. The mesh was generated automatically by specifying basic mesh parameters and constraints with subsequent refinement based on the doping distribution.

DEVEDIT3D generates two types of files: A DEVEDIT3D input file and the structure file. The first can be run in DECKBUILD to produce the corresponding structure file and is included here as the first portion of the input file. The second can be read in directly by ATLAS in the mesh statement. Note that the DevEdit3D input file can be edited as any other input file. It is straightforward to change type and value of the doping associated with each region or resize regions. More importantly, DevEdit3D input files can also be read directly into the graphical user interface of `DevEdit3D` to provide all the menu options used to construct the structure.

The ATLAS simulation begins by reading the structure from DEVEDIT3D. DECKBUILD provides an autointerface between DEVEDIT3D and ATLAS so that the structure produced by DEVEDIT3D is transferred to ATLAS without having to indicate the `mesh` statement (commented out in this example). Without the automatic DEVEDIT3D/ATLAS interface under DeckBuild, the mesh statement is needed to load the structure and the mesh.

The first active statements in the ATLAS portion of the input file are contact and models definitions. In the `contact` statement the workfunction is specified for the gate contact to reflect the properties of polysilicon. The set of physical models used in the simulation includes Shockley-Read-Hall and Auger recombination as well as the CVT mobility model accounting for doping, parallel and transverse electric field mobility dependencies. The `material` statement and it's parameters are not defined in the input file implying that the default values for silicon are used. Two-carrier solution of the problem is indicated through the parameter `carrier=2` in the `method` statement. This parameter, however, is omitted here since it represents the default setting.

In practical applications while simulating the latched state of a transistor in a memory cell, a large lumped resistor must be attached to the drain since the latter is in the floating state. This should adequately represent the MOSFET in either high or low state that may be used in the SRAM cell. This, however, lies beyond the scope of the example considered here. The purpose of the current example is to demonstrate general capabilities in simulating the SEU by analyzing the reaction of the device in the off state to the charge generation of the SEU.

The simulation is first performed to obtain the condition of the structure prior to the particle hit. This condition is with the drain voltage of 5V, all other electrodes grounded. This condition of the structure is saved in a solution file for the use in the second atlas run.

The parameters of the charge track are specified in the {bold } singleeventupset statement. The general description of the SEU physical model employed in ATLAS 3D mode as well as the description of all of the parameters for specifying position and time dependent generation is given in the Appendix to the current example description.

The first SEU simulation is performed for a normal particle incidence through the drain region (along Y direction) with a maximum carrier density generated of 1.e18 cm-3. The characteristic radius of the track was defined to be 0.05 micron, the characteristic time of the Gaussian time dependency - 2 ps, and the time instant of the peak of generation - 4 ps.

The transient simulation is performed to monitor the effects of SEU generation. The initial time steps in the solve statements are defined by the user (eg. 0.05 ps in the first solve statement), while the subsequent time steps are selected automatically. The simulation is performed up to 0.1 microsecond of physical time. To analyze the effect of carrier generation on internal physical distributions, an output structure file is saved for the instant of the peak of generation (t=4 ps).

The external terminals transient characteristics are saved in the .log file. The MOSFET transient response in the form of drain current transient characteristic is displayed using TONYPLOT.

The simulation then proceeds to the second ATLAS input file for simulating oblique particle incidence. This input file contains the same CONTACT and MODELS definitions as the previous one. The SEU definition of the particle hitting the device along its longer diagonal through the drain region (far left top corner) down to the near right bottom corner of the structure is specified in the single statement. All other parameters of the SEU are the same as in the previous ATLAS simulation. The condition of the structure of Vds=5; V is loaded and the transient response of the SEU is simulated for the second SEU conditions.

The external terminals transient characteristics are saved in the .log file and the structure information is saved in the solution file at the peak of generation.

Both drain transient current characteristics corresponding to the different SEU conditions are then displayed in TONYPLOT. The peak of the current associated with the drift collection is achieved at approximately 6 - 7 ps. Note that the peak of generation is taking place at 4 ps. The drift collection continues for about ~100 ps when the charge in the depletion region is practically collected. The peak of the current for the diagonal incidence is calculated to be slightly higher than for the normal incidence. The decrease in current is however slower for the normal incidence making the total extracted charge greater.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

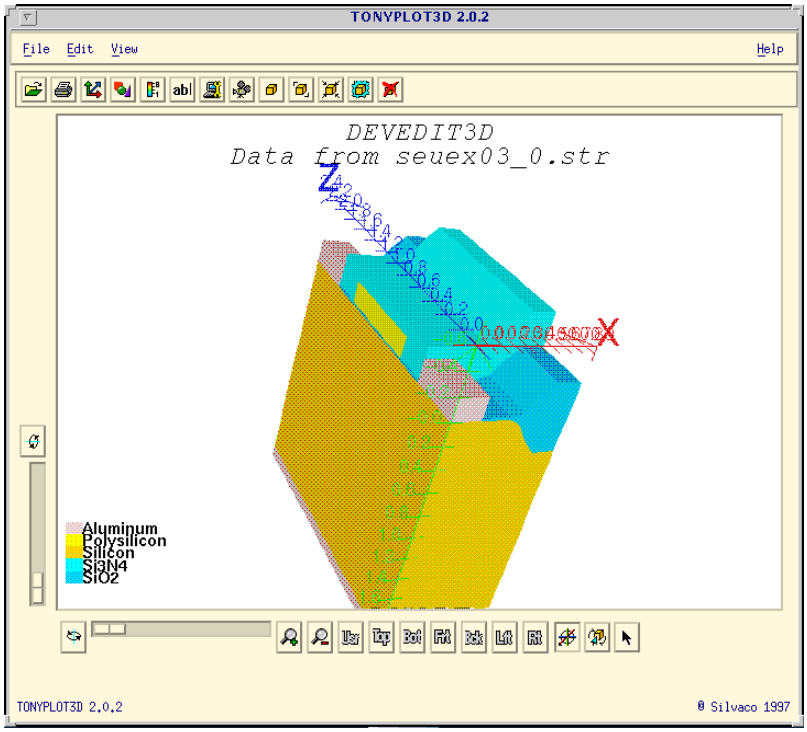


Figure 19.5: 3D MOSFET structure defined in DevEdit3D

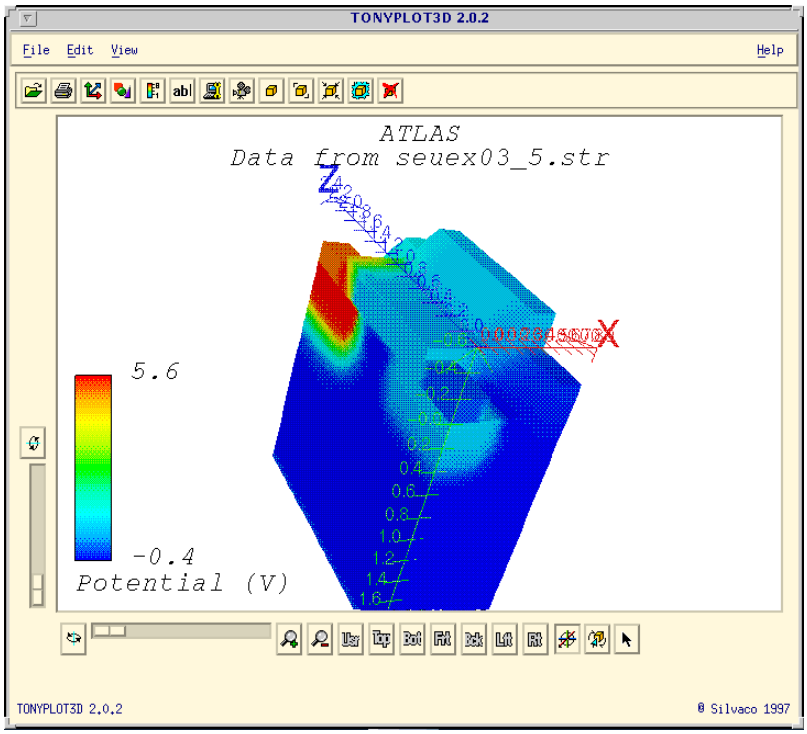


Figure 19.6: Potential in the 3D MOSFET before the SEU strike

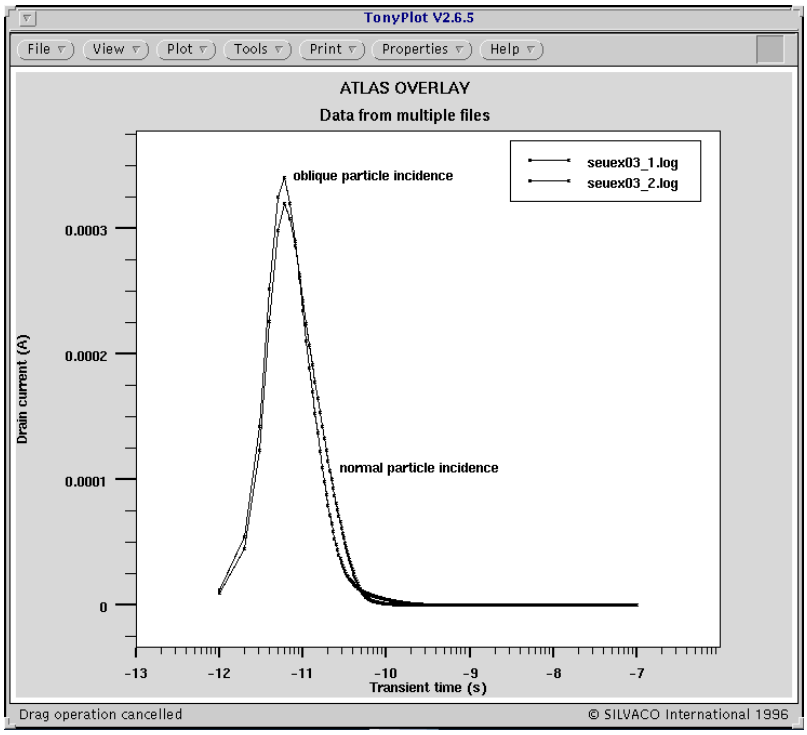


Figure 19.7: Comparison of transient current response to SEU for normal incident and oblique incident charge tracks

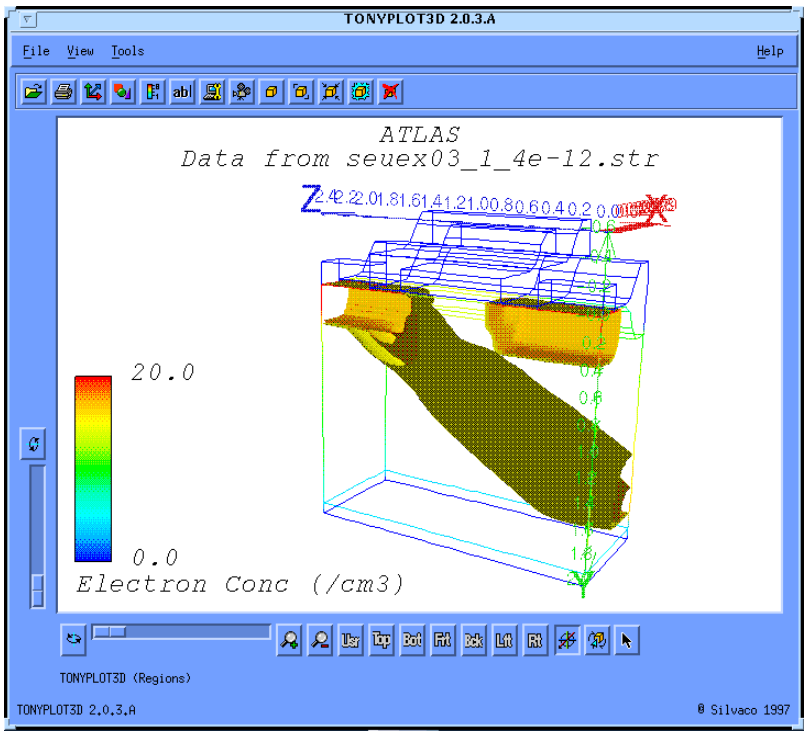


Figure 19.8: Charge track for SEU with oblique incidence. Carrier density is constant on the isosurface 3D contours

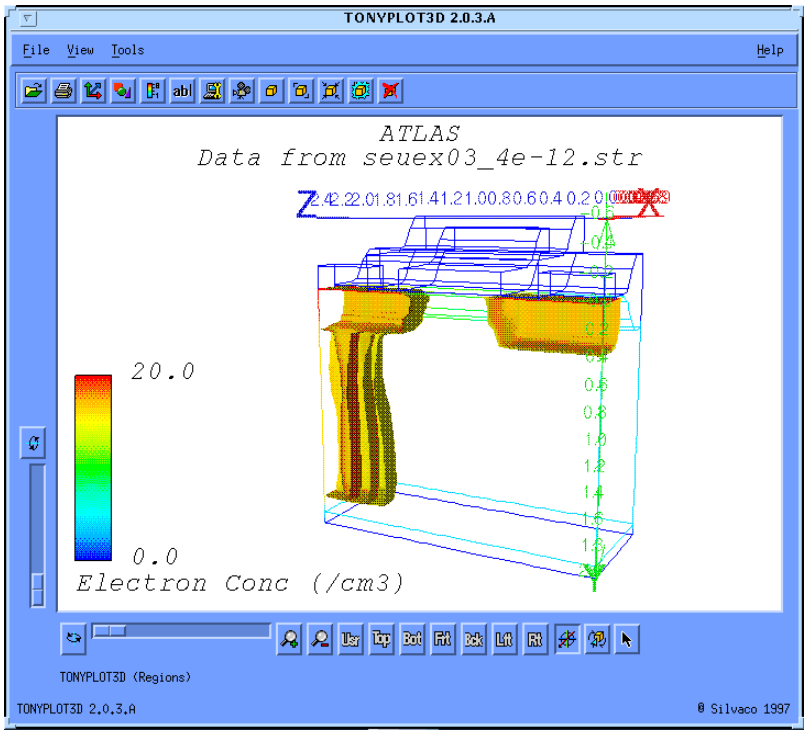


Figure 19.9: Charge track for SEU with normal incidence. Carrier density is constant on the isosurface 3D contours

Input File seu/seuex03.in:

```
1   go DevEdit
2   #
3   #  3D MOS transistor SEU simulation
4   #
5   #  SILVACO INTERNATIONAL
6
7   DevEdit version="2.0" library="1.14"
8
9   work.area left=0 top=-0.6 right=1 bottom=2.1
10
11  # SILVACO Library V1.14
12
13  region reg=1 mat=Silicon color=0xffc000 pattern=0x3 Z1=0 Z2=2.5 \
14  points="0,0 0.25,0 0.5,0 0.6,0.05 0.7,0.2 0.8,0.3 1,0.3 1,2 0,2 0,0"
15  #
16  impurity id=1 region.id=1 imp=Boron color=0x906000 \
17  x1=0 x2=0 y1=0 y2=0 \
18  peak.value=2e+16 ref.value=0 z1=0 z2=0 comb.func=Multiply \
19  rolloff.y=both conc.func.y=Constant \
20  rolloff.x=both conc.func.x=Constant \
```

```
rolloff.z=both conc.func.z=Constant
#
constr.mesh region=1 default

region reg=2 mat="Silicon Oxide" color=0xff pattern=0x1 Z1=0 Z2=2.5 \
points="0,-0.02 0.25,-0.02 0.5,-0.02 0.6,-0.05 0.7,-0.2 0.8,-0.3 1,-0.3
  1,0.3 0.8,0.3 0.7,0.2 0.6,0.05 0.5,0 0.25,0 0,0 0,-0.02"
#
constr.mesh region=2 default

region reg=3 mat="Silicon Nitride" color=0xffff pattern=0x2 Z1=0.5 Z2=2 \
points="0,-0.3 0.5,-0.3 0.6,-0.35 0.7,-0.5 0.8,-0.6 1,-0.6 1,-0.5 1,-0.3
  0.8,-0.3 0.7,-0.2 0.6,-0.05 0.5,-0.02 0.25,-0.02 0,-0.02 0,-0.2 0,-0.3"
#
constr.mesh region=3 default

region reg=4 name=gate mat=PolySilicon elec.id=1 color=0xffff00 pat-
  tern=0x4 Z1=0.8 Z2=1.7 \
points="0,-0.2 0.25,-0.2 0.5,-0.2 0.6,-0.25 0.7,-0.4 0.8,-0.5 1,-0.5 1,-
  0.3 0.8,-0.3 0.7,-0.2 0.6,-0.05 0.5,-0.02 0.25,-0.02 0,-0.02 0,-0.2"
#
constr.mesh region=4 default

region reg=5 name=source mat=Aluminum elec.id=2 color=0xffc8c8 pat-
  tern=0x6 Z1=0 Z2=0.5 \
points="0,-0.2 0.25,-0.2 0.25,-0.02 0.25,0 0,0 0,-0.02 0,-0.2"
#
constr.mesh region=5 default

region reg=6 name=drain mat=Aluminum elec.id=3 color=0xffc0c0 pattern=0x6
  Z1=2 Z2=2.5 \
points="0,-0.2 0.25,-0.2 0.25,-0.02 0.25,0 0,0 0,-0.02 0,-0.2"
#
constr.mesh region=6 default

region reg=7 name=substrate mat=Aluminum elec.id=4 color=0xffc8c8 pat-
  tern=0x7 Z1=0 Z2=2.5 \
points="0,2 1,2 1,2.1 0,2.1 0,2"
#
constr.mesh region=7 default


impurity id=1 imp=Phosphorus color=0x906000 \
x1=0 x2=0.3 y1=0 y2=0 \
peak.value=1e+20 ref.value=0 z1=0 z2=0.8 comb.func=Multiply \
```

```
rolloff.y=both conc.func.y=Gaussian conc.param.y=0.1 \
rolloff.x=both conc.func.x="Error Function" conc.param.x=0.05 \
rolloff.z=both conc.func.z="Error Function" conc.param.z=0.05
impurity id=2 imp=Phosphorus color=0x906000 \
x1=0 x2=0.3 y1=0 y2=0 \
peak.value=1e+20 ref.value=0 z1=1.7 z2=2.5 comb.func=Multiply \
rolloff.y=both conc.func.y=Gaussian conc.param.y=0.1 \
rolloff.x=both conc.func.x="Error Function" conc.param.x=0.05 \
rolloff.z=both conc.func.z="Error Function" conc.param.z=0.05
impurity id=3 imp=Boron color=0x906000 \
x1=0 x2=0.5 y1=0.05 y2=0.05 \
peak.value=1e+17 ref.value=0 z1=0 z2=2.5 comb.func=Multiply \
rolloff.y=both conc.func.y=Gaussian conc.param.y=0.05 \
rolloff.x=both conc.func.x="Step Function" conc.param.x=0.05 \
rolloff.z=both conc.func.z="Error Function" conc.param.z=0.05
impurity id=4 imp=Boron color=0x906000 \
x1=0.5 x2=1 y1=0.4 y2=0.4 \
peak.value=1e+17 ref.value=0 z1=0 z2=2.5 comb.func=Multiply \
rolloff.y=both conc.func.y=Gaussian conc.param.y=0.05 \
rolloff.x=both conc.func.x="Step Function" conc.param.x=0.05 \
rolloff.z=both conc.func.z="Error Function" conc.param.z=0.05

# Set Meshing Parameters
#
base.mesh height=0.2 width=0.1
#
bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=0.001
  line.straightening=1 align.points when=automatic
#
imp.refine imp="Phosphorus" sensitivity=1
imp.refine min.spacing=0.01 z=0
#
constr.mesh max.angle=90 max.ratio=300 max.height=1 \
max.width=1 min.height=0.0001 min.width=0.0001
#
constr.mesh type=Semiconductor default
#
constr.mesh type=Insulator default
#
constr.mesh type=Metal default
#
constr.mesh type=Other default
#
```

```
constr.mesh region=1 default
#
constr.mesh region=2 default
#
constr.mesh region=3 default
#
constr.mesh region=4 default
#
constr.mesh region=5 default
#
constr.mesh region=6 default
#
constr.mesh region=7 default
#
# Perform mesh operations
#
Mesh Mode=MeshBuild
Refine Mode=Y P1=0.012639146567718,-0.016114121268342
  P2=0.492926716141002,0.185606236296171
Refine Mode=Y P1=0.013788159892056,0.00988889357395856
  P2=0.490628689492326,0.0705625948726598

imp.refine imp="Phosphorus" sensitivity=1
imp.refine min.spacing=0.01 z=0

constr.mesh max.angle=90 max.ratio=300 max.height=1 \
max.width=1 min.height=0.0001 min.width=0.0001
#
constr.mesh type=Semiconductor default
#
constr.mesh type=Insulator default
#
constr.mesh type=Metal default
#
constr.mesh type=Other default

z.plane z=0 spacing=0.25
#
z.plane z=0.5 spacing=0.1
#
z.plane z=0.8 spacing=0.1
#
z.plane z=1.7 spacing=0.08
#
```

```
143 z.plane z=2 spacing=0.1
144 #
145 z.plane z=2.5 spacing=0.25
146 #
147 z.plane max.spacing=1000000 max.ratio=1.5
148
149 base.mesh height=0.2 width=0.1
150
151 bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=0.001
      line.straightening=1 align.Points when=automatic
152
153
154 structure outf=seuex03_0.str
155
156 go atlas
157
158 title   SEU of MOSFET: normal incidence
159
160 # Load the structure generated in DevEdit...
161 # While running in the deckbuild after DevEdit Input File the structure is
162 # transfered automatically: no mesh statement is needed.
163
164
165 # Gate contact definition
166 contact num=1 name=gate workfun=4.1
167
168 # Models specification
169 models   srh auger cvt
170
171
172 # Single Event Upset
173 # Specify the charge track: normal incidence through the drain
174
175 single entry="0.1,0.0,2.25" exit="0.1,2.0,2.25" radius=0.05 density=1.e18
      \
176        t0=4.e-12 tc=2.e-12
177
178
179 # Specify numerical method
180 method newton
181
182 # Get initial condition of the structure
183 solve init
184
```

```
# Apply voltage to the drain
# Get condition of the structure prior to the particle hit

solve vdrain=0.025
solve vdrain=0.1
solve vdrain=0.2
solve vdrain=0.5 vstep=0.5 vfinal=5 name=drain

save outf=seuex03_5.str

# SEU: Calculate transient response
method halfimpl dt.min=1.e-12

log outf=seuex03_1.log
solve tfinal=4.e-12 tstep=1.e-12
save outf=seuex03_4e-12.str
solve tfinal=1.e-7  tstep=1.e-12


#

go atlas

title SEU of MOSFET: oblique incidence

mesh infile=seuex03_0.str master.in

# Gate contact definition
contact num=1 name=gate workfun=4.1

# Models specification
models   srh auger cvt

# Single Event Upset
# Specify the charge track: along drain-substrate long diagonal

single entry="0.1,0.0,2.25" exit="1.0,2.0,0.0" radius=0.05 density=1.e18
   \
       t0=4.e-12 tc=2.e-12

# Specify numerical method
method newton

```

```
# Load previously obtained solution VD=5V

load inf=seuex03_5.str

solve vdrain=5.0

# SEU: Calculate transient response

method halfimpl dt.min=1.e-12

log outf=seuex03_2.log

solve tfinal=4.e-12 tstep=1.e-12

save outf=seuex03_1_4e-12.str

solve tfinal=1.e-7  tstep=1.e-12

# Plot drain current transients for the normal and oblique particle inci-
  dence
tonyplot -overlay seuex03_1.log seuex03_2.log -set seuex03_log.set
#

quit


```

19.1.4 seuex04.in: Arbitrary Charge Generation using C-Interpreter

Requires: DEVICE3D, C-INTERPRETER

This examples demonstrates the use of the C interpreter function `F3.RADIATE`. This function allows the user to specify arbitrary electron hole pair generation as a function of f position and time. This example illustrates the functionality using a simple block function for the charge. Other template files are loaded along with the input file. These can be substituted in the line:

```
beam num=1 f3.radiate=<filename>
```

Note that a generation value is calculated by the C function for each XYZ location in the device. This value is multiplied by the value of the `b1` of the `solve` statement before being applied.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

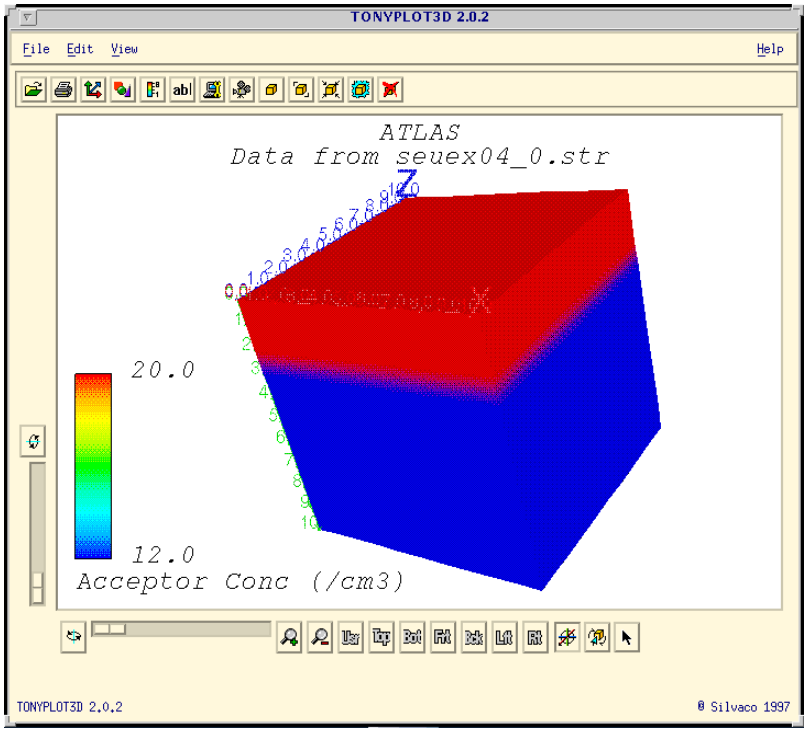


Figure 19.10: Doping in a 3D diode structure defined using ATLAS syntax

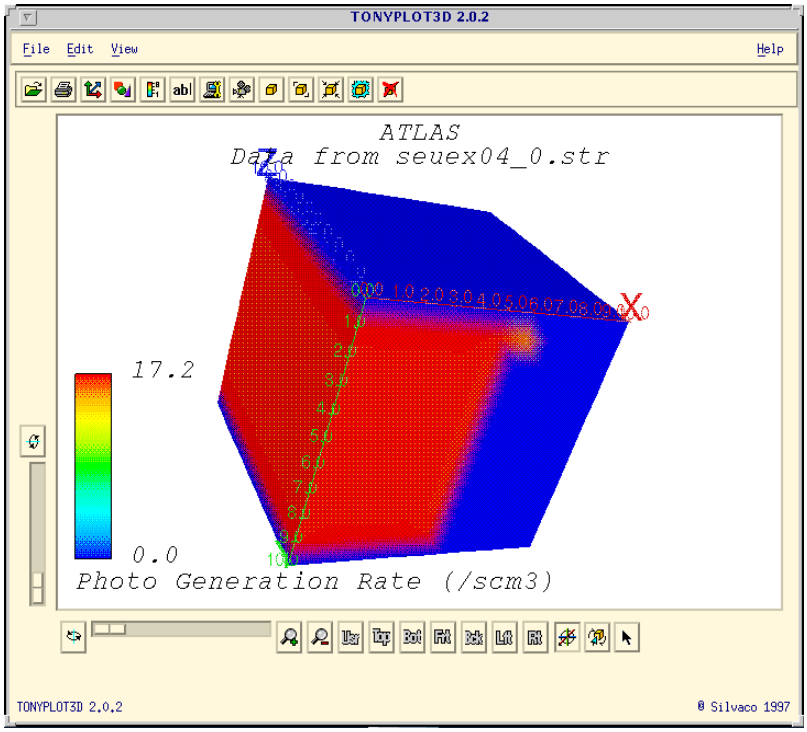


Figure 19.11: Charge generation rate user-defined using a C-Interpreter function

Input File seu/seuex04.in:

```
1    go atlas
```

```
2
3   mesh three.d
4   x.m loc=0 spac=1
5   x.m loc=10 spac=1
6
7   y.m loc=0 spac=1
8   y.m loc=10 spac=1
9
10  z.m loc=0 spac=1
11  z.m loc=10 spac=1
12
13  region num=1 silicon
14
15  electrode name=anode top
16  electrode name=cathode bottom
17
18  doping uniform n.type conc=1e15
19  doping uniform p.type conc=1e20 y.bot=2.1
20
21  # The radiation function is specified here
22
23  beam f3.radiate=seuex04_0.lib
24
25  solve init
26  # setting the intensity of the "beam" turns on the
27  # pair generation function specified above.
28
29  solve b1=1
30  save outf=seuex04_0.str
31
32  quit
33
```

19.1.5 seuex05.in: Defining a Sphere of Charge using C-Interpreter

Requires: DEVICE3D, C-INTERPRETER

This examples demonstrates the **f3.radiate** function by generating a sphere of charge inside a cube of semiconductor material.

Note that a generation value calculated by the C function for each XYZ location in the device. This value is multiplied by the value of the b1 of the solve statement before being applied.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

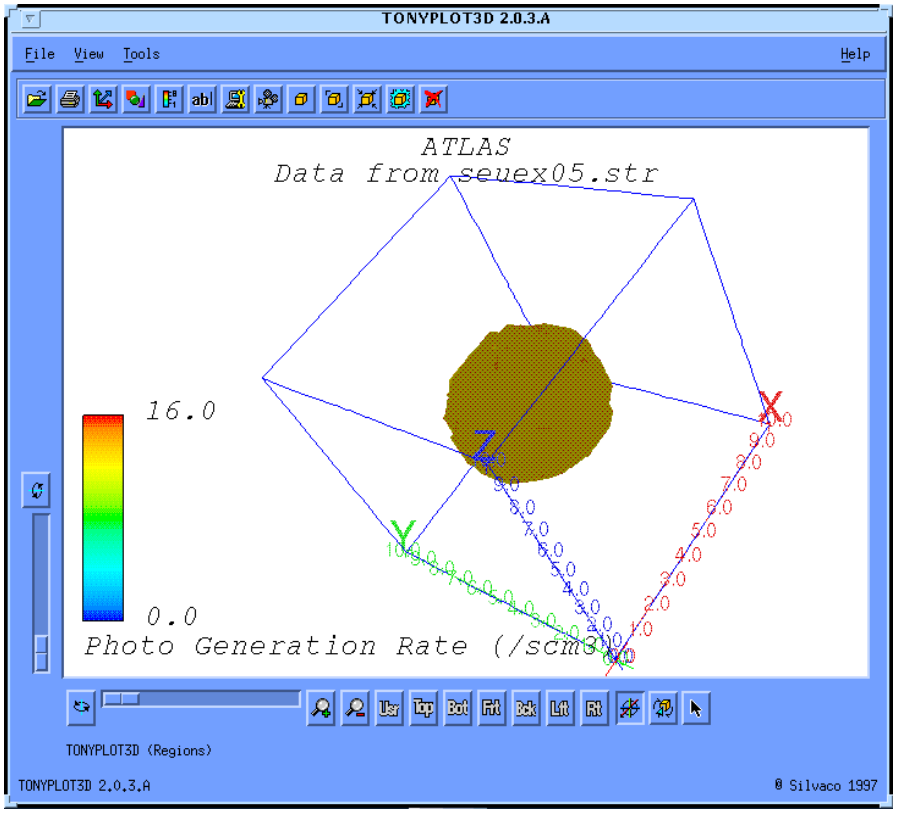


Figure 19.12: Sphere of charge displayed as 3D isocontours

Input File seu/seuex05.in:

```
1   go atlas
2   Title Radiation interpreter function example
3   #
4   # Radiation interpreter function example
5   # SILVACO International 1996
6   #
7   # SECTION 1: Mesh Specification
8   #
9   mesh three.d space.mult=1.0
10  #
11  x.mesh loc=0.0  spacing=1.0
12  x.mesh loc=10.0 spacing=1.0
13  #
14  y.mesh loc=0.0   spacing=0.1
15  y.mesh loc=5.0   spacing=0.2
16  y.mesh loc=10.0  spacing=0.1
17  #
18  z.m  l=0    spacing=1
19  z.m  l=5.0 s=1
```

```
20 z.m  l=10.0 spacing=1
21 #
22 # SECTION 2: Structure Specification
23 #
24 region num=1 material=Silicon
25 #
26 elec num=1 name=anode x.min=0.0 x.max=10.0 y.max=0.0
27 elec num=2 name=cathode bottom
28 #
29 doping uniform conc=1e14 n.type
30 doping gaus peak=0.0 char=0.1 conc=1e18 p.type dir=y
31 doping gaus peak=10.0 char=0.1 conc=1e18 n.type dir=y
32 #
33 # SECTION 3: Material Model Specification
34 #
35 material taup0=2.e-6 taun0=2.e-6
36 models srh auger conmob fldmob
37 #
38 # define a function for the radiation
39 beam  f3.radiate=seuex05.lib
40 #
41 # SECTION 4: Initial Solution
42 #
43
44
45 solve init
46 #
47 # SECTION 5: Current vs. intensity
48 #
49 method newton
50 solve b1=0.000001 outf=seuex05.str master
51 quit
```

19.1.6 seuex06.in: SEU in Memory Cell Transistor

Requires: DevEdit3D/DEVICE3D

This example demonstrates SEU in a MOSFET with an external resistor and capacitor emulating a RAM cell. The example shows:

- formation of structure in DEVEDIT3D
- parameterized structural and doping parameters
- automatic interface from DEVEDIT3D to DEVICE3D
- setting of external resistor and capacitor on the `drain` contact
- definition of the SEU pulse

- DC simulation to bias MOSFET in initial cell condition
- transient simulation of SEU strike
- analysis of voltage drop at the drain contact

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

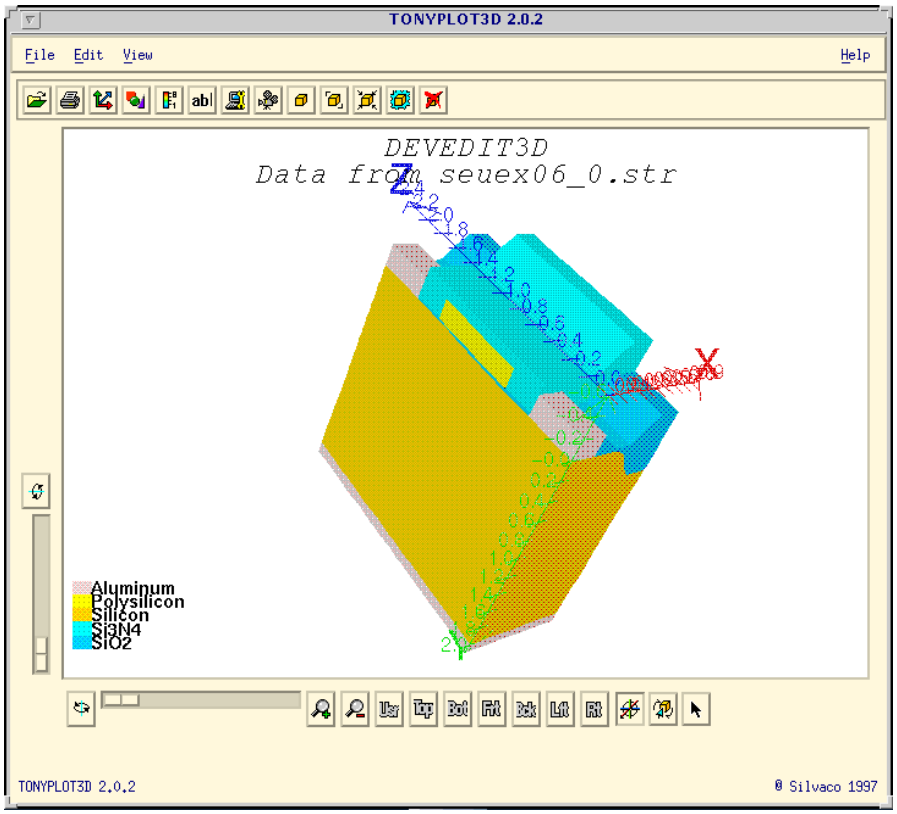


Figure 19.13: 3D MOSFET defined using DevEdit3D. External elements are used to complete the cell

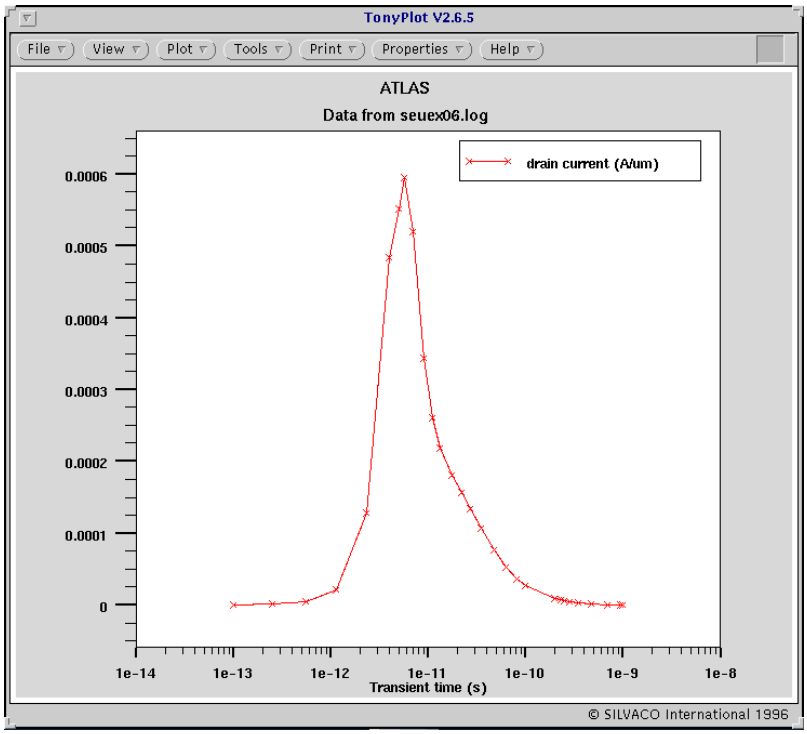


Figure 19.14: Current pulse from an SEU Pulse in through the MOS drain

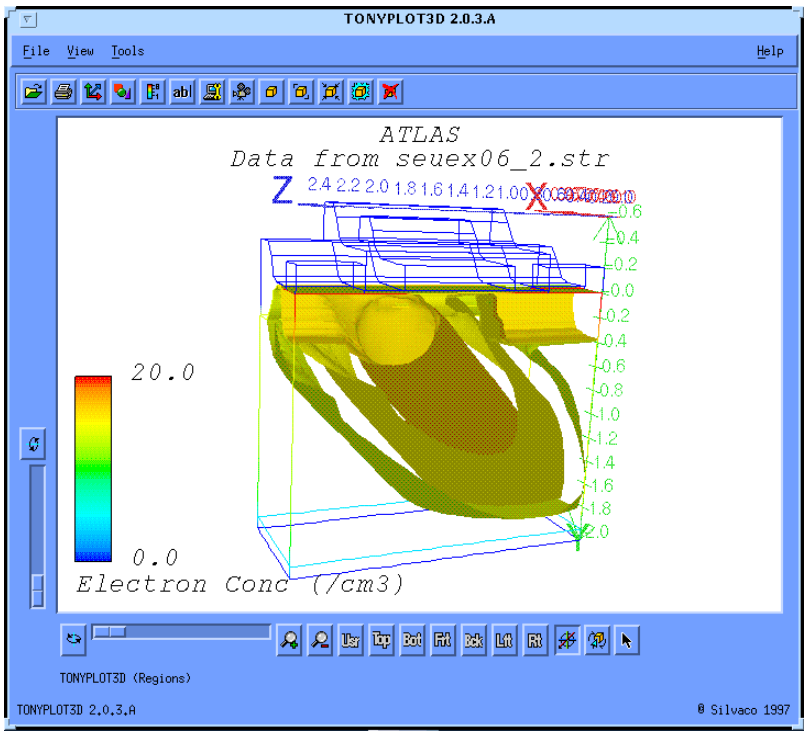


Figure 19.15: 3D contours of electron concentration showing the track of the SEU

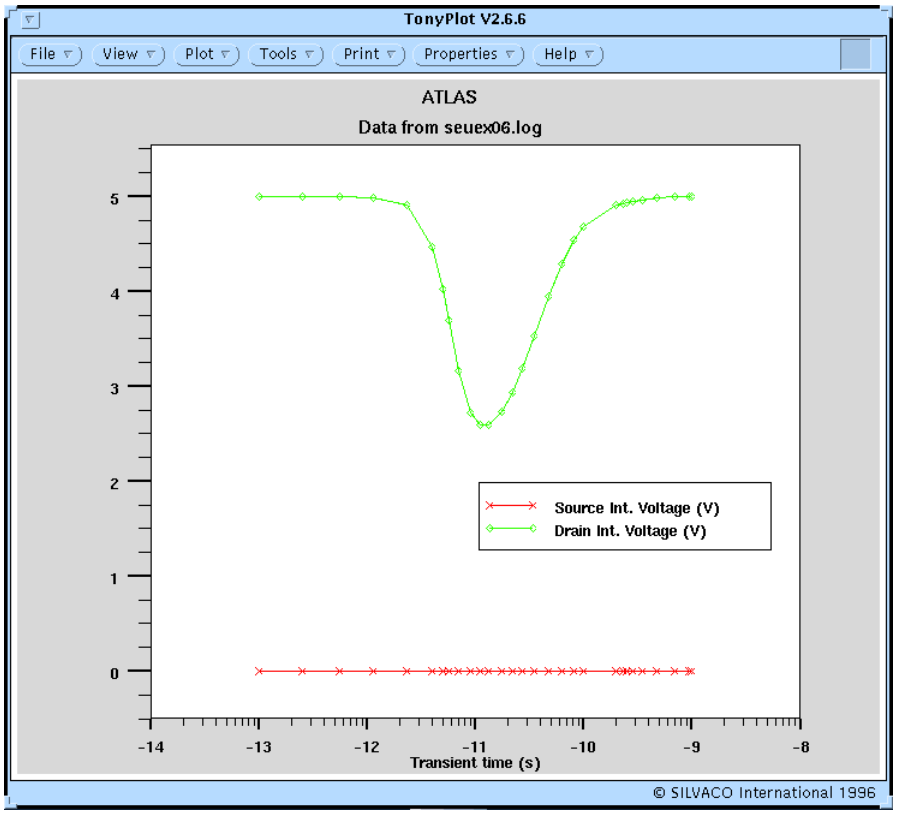


Figure 19.16: Drain voltage during the SEU pulse. The drain voltage is pulled back to the supply by the external resistance

Input File seu/seuex06.in:

```
1   go DevEdit
2
3   set pkfld = 1.0e17
4
5   set pkchan = 5.0e16
6
7   set width=1.0
8
9   set len=0.5-($width/2)
10
11
12  go DevEdit
13  #
14  #  3D MOS transistor
15  #  Steady-state simulation
16  #
17  #  SILVACO INTERNATIONAL
18
19  DevEdit version="2.0" library="1.14"
```

```
20
21 work.area left=$"len" top=-0.6 right=1 bottom=2.1
22
23 # SILVACO Library V1.14
24
25 region reg=1 mat=Silicon color=0xffc000 pattern=0x3 Z1=0 Z2=2.5 \
26 points="$"len",0 0.25,0 0.5,0 0.6,0.05 0.7,0.2 0.8,0.3 1,0.3 1,2 $"len",2
     $"len",0"
27 #
28 impurity id=1 region.id=1 imp=Boron color=0x906000 \
29 x1=$"len" x2=0 y1=0 y2=0 \
30 peak.value=2e+16 ref.value=0 z1=0 z2=0 comb.func=Multiply \
31 rolloff.y=both conc.func.y=Constant \
32 rolloff.x=both conc.func.x=Constant \
33 rolloff.z=both conc.func.z=Constant
34 #
35 constr.mesh region=1 default
36
37 region reg=2 mat="Silicon Oxide" color=0xff pattern=0x1 Z1=0 Z2=2.5 \
38 points="$"len",-0.02 0.25,-0.02 0.5,-0.02 0.6,-0.05 0.7,-0.2 0.8,-0.3 1,-
     0.3 1,0.3 0.8,0.3 0.7,0.2 0.6,0.05 0.5,0 0.25,0 $"len",0 $"len",-0.02"
39 #
40 constr.mesh region=2 default
41
42 region reg=3 mat="Silicon Nitride" color=0xffff pattern=0x2 Z1=0.5 Z2=2 \
43 points="$"len",-0.3 0.5,-0.3 0.6,-0.35 0.7,-0.5 0.8,-0.6 1,-0.6 1,-0.5
     1,-0.3 0.8,-0.3 0.7,-0.2 0.6,-0.05 0.5,-0.02 0.25,-0.02 $"len",-0.02
     $"len",-0.2 $"len",-0.3"
44 #
45 constr.mesh region=3 default
46
47 region reg=4 name=gate mat=PolySilicon elec.id=1 color=0xffff00 pat-
     tern=0x4 Z1=0.8 Z2=1.7 \
48 points="$"len",-0.2 0.25,-0.2 0.5,-0.2 0.6,-0.25 0.7,-0.4 0.8,-0.5 1,-0.5
     1,-0.3 0.8,-0.3 0.7,-0.2 0.6,-0.05 0.5,-0.02 0.25,-0.02 $"len",-0.02
     $"len",-0.2"
49 #
50 constr.mesh region=4 default
51
52 region reg=5 name=source mat=Aluminum elec.id=2 color=0xffc8c8 pat-
     tern=0x6 Z1=0 Z2=0.5 \
53 points="$"len",-0.2 0.25,-0.2 0.25,-0.02 0.25,0 $"len",0 $"len",-0.02
     $"len",-0.2"
54 #
55 constr.mesh region=5 default
56
```

```
region reg=6 name=drain mat=Aluminum elec.id=3 color=0xffc0c0 pattern=0x6
  Z1=2 Z2=2.5 \
points="$"len",-0.2 0.25,-0.2 0.25,-0.02 0.25,0 $"len",0 $"len",-0.02
  $"len",-0.2"
#
constr.mesh region=6 default

region reg=7 name=substrate mat=Aluminum elec.id=4 color=0xffc8c8 pat-
  tern=0x7 Z1=0 Z2=2.5 \
points="$"len",2 1,2 1,2.1 0,2.1 $"len",2"
#
constr.mesh region=7 default


impurity id=1 imp=Phosphorus color=0x906000 \
x1=$"len" x2=0.3 y1=0 y2=0 \
peak.value=1e+20 ref.value=0 z1=0 z2=0.8 comb.func=Multiply \
rolloff.y=both conc.func.y=Gaussian conc.param.y=0.1 \
rolloff.x=both conc.func.x="Error Function" conc.param.x=0.05 \
rolloff.z=both conc.func.z="Error Function" conc.param.z=0.05
impurity id=2 imp=Phosphorus color=0x906000 \
x1=$"len" x2=0.3 y1=0 y2=0 \
peak.value=1e+20 ref.value=0 z1=1.7 z2=2.5 comb.func=Multiply \
rolloff.y=both conc.func.y=Gaussian conc.param.y=0.1 \
rolloff.x=both conc.func.x="Error Function" conc.param.x=0.05 \
rolloff.z=both conc.func.z="Error Function" conc.param.z=0.05
impurity id=3 imp=Boron color=0x906000 \
x1=$"len" x2=0.5 y1=0.05 y2=0.05 \
peak.value=$pkchan ref.value=0 z1=0 z2=2.5 comb.func=Multiply \
rolloff.y=both conc.func.y=Gaussian conc.param.y=0.05 \
rolloff.x=both conc.func.x=Gaussian conc.param.x=0.05 \
rolloff.z=both conc.func.z="Error Function" conc.param.z=0.05
impurity id=4 imp=Boron color=0x906000 \
x1=0.5 x2=1 y1=0.3 y2=0.3 \
peak.value=$pkfld ref.value=0 z1=0 z2=2.5 comb.func=Multiply \
rolloff.y=both conc.func.y=Gaussian conc.param.y=0.25 \
rolloff.x=both conc.func.x=Gaussian conc.param.x=0.25 \
rolloff.z=both conc.func.z="Error Function" conc.param.z=0.05

# Set Meshing Parameters
#
base.mesh height=0.2 width=0.1
#
```

```
97  bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=0.001
      line.straightening=1 align.points when=automatic
98  #
99  imp.refine imp="Phosphorus" sensitivity=1
100 imp.refine min.spacing=0.01 z=0
101 #
102 constr.mesh max.angle=90 max.ratio=300 max.height=1 \
103 max.width=1 min.height=0.0001 min.width=0.0001
104 #
105 constr.mesh type=Semiconductor default
106 #
107 constr.mesh type=Insulator default
108 #
109 constr.mesh type=Metal default
110 #
111 constr.mesh type=Other default
112 #
113 constr.mesh region=1 default
114 #
115 constr.mesh region=2 default
116 #
117 constr.mesh region=3 default
118 #
119 constr.mesh region=4 default
120 #
121 constr.mesh region=5 default
122 #
123 constr.mesh region=6 default
124 #
125 constr.mesh region=7 default
126 #
127 # Perform mesh operations
128 #
129 Mesh Mode=MeshBuild
130 Refine Mode=Y P1=$"len",-0.016114121268342
      P2=0.492926716141002,0.185606236296171
131 Refine Mode=Y P1=$"len",0.00988889357395856
      P2=0.490628689492326,0.0705625948726598
132
133 imp.refine imp="Phosphorus" sensitivity=1
134 imp.refine min.spacing=0.01 z=0
135
136 constr.mesh max.angle=90 max.ratio=300 max.height=1 \
137 max.width=1 min.height=0.0001 min.width=0.0001
```

```
#
constr.mesh type=Semiconductor default
#
constr.mesh type=Insulator default
#
constr.mesh type=Metal default
#
constr.mesh type=Other default

z.plane z=0 spacing=0.25
#
z.plane z=0.5 spacing=0.1
#
z.plane z=0.8 spacing=0.1
#
z.plane z=1.7 spacing=0.08
#
z.plane z=2 spacing=0.1
#
z.plane z=2.5 spacing=0.25
#
z.plane max.spacing=1000000 max.ratio=1.5

base.mesh height=0.2 width=0.1

bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=0.001
      line.straightening=1 align.Points when=automatic

structure outf=seuex06_0.str

go atlas
#
contact name=gate workfun=4.17
contact name=drain res=1.0e4 cap=1.0e-15
#
models srh auger cvt
#
# Single Event Upset
# Specify the charge track: along drain-substrate long diagonal

single entry="0.0,-0.6,2.5" exit="1.0,2.0,0.0" radius=0.1 density=1.e18 \
        t0=4.e-12 tc=2.e-12

```

```
# Specify numerical method
method newton carr=2

solve vdrain=0.1
solve vdrain=1.0 vstep=1.0 vfinal=5.0 name=drain
save outf=seuex06_1.str

#for subsequent runs can save time by using LOAD
#load inf=seuex06_1.str master
#solve prev

#  Calculate transient response

method newton dt.min=1.e-15

log outf=seuex06.log

solve tfinal=4.e-12 tstep=1.e-13

save outf=seuex06_2.str

solve tfinal=1.e-10  tstep=1.e-12

save outf=seuex06_3.str

solve tfinal=1.e-9  tstep=1.e-10
```

This page is intentionally left blank

20.1. INTERCONNECT: Interconnect Parasitics Application Examples

20.1.1 interconnectex01.in: 2D Inter-Track Capacitance using ELITE

Requires: SSUPREM4/ELITE/SPISCES

This examples shows the use of ATHENA and ATLAS to extract 2D inter-metal interconnect capacitances.

- a simple LOCOS oxidation is simulated in SSUPREM4 to create a non-planar surface
- two metal tracks are defined
- inter-metal dielectric is deposited using ELITE
- the capacitance between each pair of metal tracks is calculated.

The input file starts by specifying the process flow for this example. The first step is a field oxidation on one side of the structure to create a non-planar surface. Metal is deposited using a simple conformal deposition in ATHENA and then patterned. The inter-metal dielectric is made of a nitride-oxide sandwich to demonstrate the capability of ATLAS to handle insulators of different permittivities.

The oxide deposition is done using ELITE to provide realistic topology for the capacitance extraction. For details on the syntax used by ELITE, see the ATHENA_ELITE examples section.

The `electrode` statement is used at the end of the ATHENA run to define the metallic contacts between which the capacitance will be taken.

In ATLAS the `ac` parameter of the `solve` statement is used to activate the small signal AC analysis. A frequency of 1MHz is used. A CV curve is formed by ramping one conductor to 2.0V. The capacitance versus bias data is stored to the log file.

Plots of the final processed structure, the C-V curve and the potential distribution of the last bias point solved can be seen in TONYPLOT.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

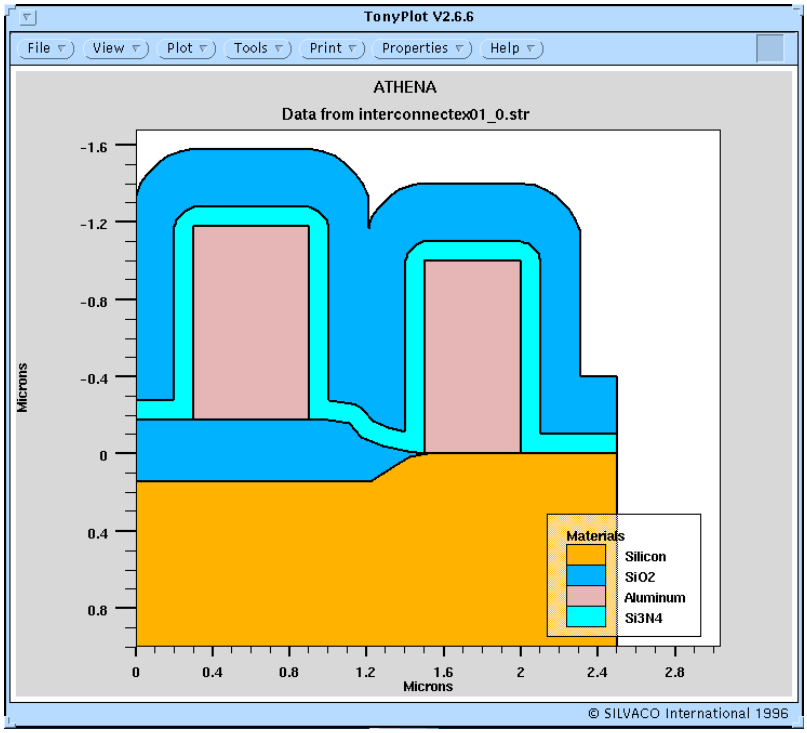


Figure 20.1: 2D Interconnect structure modeled using ATHENA

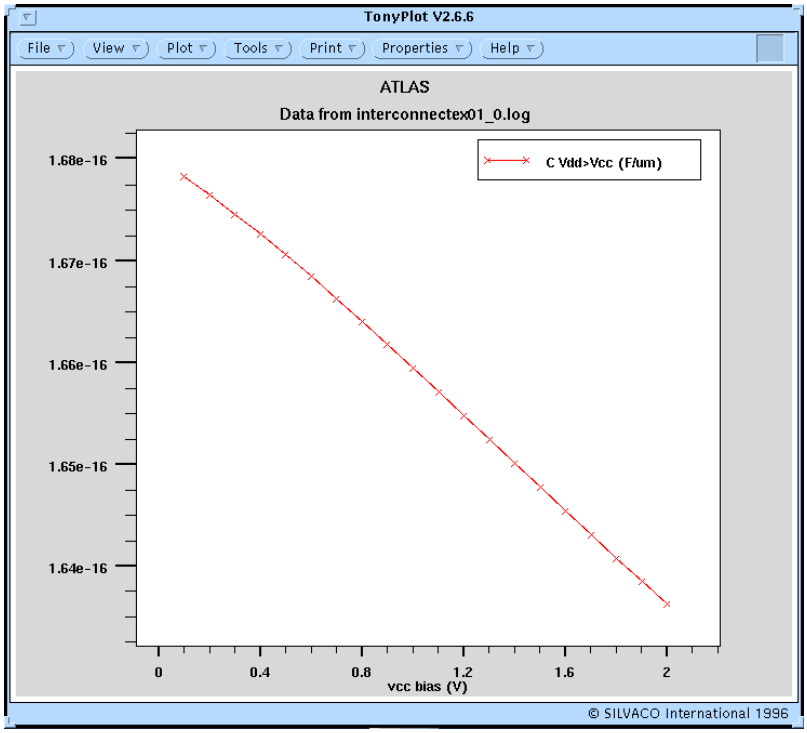


Figure 20.2: Interconnect Capacitance extracted in ATLAS. A minor bias dependence is seen from substrate depletion

Input File interconnect/interconnectex01.in:

```
1    #Inter-metal interconnect capacitance extraction
```

```
go athena
#
line x loc=0.00 spac=0.10
line x loc=2.5 spac=0.10
#
line y loc=0.00 spac=0.025
line y loc=1 spac=0.1
#
init orientation=100 c.boron=1e16

deposit nitride thick=0.20 divisions=5

etch nitride left p1.x=1.20

method fermi compress
diffus time=40 temp=1000 weto2 press=1.00 hcl.pc=0

etch nitride all

implant phosph dose=1e13 energy=30

deposit aluminum thick=1.0 divisions=8

# Etch the Aluminium...
etch alum left p1.x=0.30
etch alum right p1.x=2.00
etch alum start x=0.9 y=-3
etch cont x=1.5 y=-3
etch cont x=1.5 y=0
etch done x=0.9 y=0

deposit nitride thick=0.1 divi=3
#
# deposit oxide with step coverage of 70%
rate.depo mach=de1 cvd oxide step.cov=0.7 dep.rate=0.1 u.m

deposit mach=de1 time=3 minute div=6
#
# Name the electrodes for use in atlas2....
electrode name=vcc x=0.5 y=-0.5
electrode name=vdd x=1.7 y=-0.5

structure outfile=interconnectex01_0.str
```

```
45 tonyplot  interconnectex01_0.str -set interconnectex01_0.set
46 # now move into Atlas for capacitance extraction....
47
48 go atlas
49
50 models print
51
52 solve init
53
54
55 method newton trap
56
57 solve  prev
58
59 log outf=interconnectex01_0.log master
60 solve vstep=0.1 vfinal=2 name=vcc ac freq=1e6 term=1
61
62 extract name="cap test" max(c."vdd""vcc")
63
64 tonyplot  interconnectex01_0.log -set interconnectex01_1.set
65
66 quit
```

20.1.2 interconnectex02.in: Parallel Plate Capacitance Extraction

Requires: DEVEDIT3D/INTERCONNECT3D

This example demonstrates the extraction of capacitance from a simple 3D parallel plate structure defined using DEVEDIT3D. It shows:

- 3D structure formation in DEVEDIT3D.
- DEVEDIT3D to ATLAS interface for 3D structures.
- 3D capacitance extraction in ATLAS.

This example file consists of two parts. The first part is a DEVEDIT3D input run which forms the parallel plate structure. The second part runs an ATLAS simulation in 3D for capacitance extraction. The `go atlas` statement is the interface between these two parts.

The structure was made in DEVEDIT3D originally using the graphical mode and saved as a command file. Users should consult the DEVEDIT3D manual for details of the syntax of this input file. The `z.plane` statement is used in DEVEDIT3D

The ATLAS syntax required to read in this 3D structure and to extract the capacitance between the metal layers is simple. The automatic interface in DECKBUILD means the `go atlas` command is all that is needed to load in the 3D structure and enter the 3D portion of ATLAS. The statement:

```
models ex.cap
```

selects the interconnect capacitance extraction. The electrodes are set in DevEdit3D during the region definition and the capacitance between each pair of electrodes is returned.

The output of the example is a run-time list of the capacitances, a SPICE sub-circuit netlist of the capacitances in the structure and a 3D solution file that can be displayed in TonyPlot3D.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

20.1.3 interconnectex03.in: Multi-Level Track Capacitances using DevEdit3D

Requires: DEVEDIT3D/INTERCONNECT3D

This example demonstrates the extraction of capacitance from a two-level polysilicon and metal structure defined using DEVEDIT3D. It shows:

- 3D structure formation in DEVEDIT3D
- DEVEDIT3D to ATLAS interface for 3D structures.
- 3D capacitance extraction in ATLAS.

The structure used in this example is a common six line problem used to characterize cross-over capacitances between two interconnect layers. Three lines are specified in the XZ plane in the lower interconnect layer and three lines are specified in the XY plane in the upper interconnect layer. The capacitance between the middle conductor in the lower and upper interconnect layers corresponds to the typical cross-over capacitance found in integrated circuits. The other four lines are required to give the correct shadowing and coupling effects of the nearest neighbors.

This structure is specified using the DEVEDIT3D syntax. It was originally made graphically and the commands saved to a DEVEDIT3D command file. The ATLAS interface, commands to get the capacitance extraction activated and output are as described in the previous example.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

20.1.4 interconnectex04.in: Diffusion Resistance Extraction

Requires: DEVEDIT3D/INTERCONNECT3D

This example demonstrates the extraction of conductance (or resistance) from a 3D diffused resistor structure defined using DEVEDIT3D. It shows:

- 3D structure formation in DEVEDIT3D
- DEVEDIT3D to ATLAS interface for 3D structures.
- 3D resistance extraction in ATLAS.

This example demonstrates the extraction of the resistance of a 3D diffused resistor in a very resistive substrate. The structure is defined as an I-shaped resistor in the XZ plane with a certain depth in Y. This is embedded in a block of resistive material. The structure is defined using DEVEDIT3D. Initially the structure was described in the graphical mode of DEVEDIT3D. A command file was saved from DEVEDIT3D to reproduce this device using DECKBUILD. Two aluminum regions are specified to act as the two electrodes for this structure. The ATLAS simulation will measure the resistance between these two contacts.

The interface to ATLAS is done using the `go atlas` statement. In order to extract the resistance of the region, the only command required is `models ex.res`. The conductivity of each region is defined on the material statement. Although here this is done to emulate regions in silicon, other applications of this feature is to specify and extract resistances from purely metallic structures such as vias and complex contact and interconnect structures.

The output is produced as a SPICE sub-circuit netlist for the resistance between the two electrodes. Also a solution file is saved that can be viewed in TONYPLOT3D.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

20.1.5 interconnectex05.in: Multi-Level Metal Capacitances using ATHENA

Requires: SSUPREM4/ELITE/DEVEDIT3D/INTERCONNECT3D

This example demonstrates the extraction of parasitic capacitances between conductors in a multi-level interconnect structure. The three stages of this example show:

- formation of a 2D multi-level interconnect structure using SSUPREM4 and ELITE
- conversion of the 2D structure to 3D using DEVEDIT3D
- extraction of 3D capacitances using INTERCONNECT3D

The first stage of the input file uses ATHENA to create a structure from two metal layers, one poly layer and dielectrics. The final 3D structure will be a cross-over of Metal2 running in the X-direction. This will cross three Metal1 tracks running in the Z-direction. Two additional poly tracks are also included. These run in the Z-direction too. The 2D structure is created in the XY plane. The extent of each region into the Z direction is determined later on in the DEVEDIT3D portion of the example.

The process simulation uses ELITE to create a realistic topography and SSUPREM4 to provide a realistic oxidation model. The process sequence is:

- formation of a field oxide bird's beak using SSUPREM4
- deposition and patterning of poly
- deposition of an inter-level dielectric using ELITE
- reflow of the dielectric
- deposition and patterning of Metal1
- deposition of an inter-level dielectric using ELITE
- deposition of Metal2

The ELITE deposition of the dielectric consists of two steps. First a *machine* must be defined. A deposition *machine* is defined in terms of it's characteristic type, here we use hemispherical, and the deposition rate parameters associated with this type. The parameter, `sigma.dep` controls the surface diffusion of the deposited film. A higher value increases the surface diffusion and leads to smoother depositions.

The second stage is to run the deposition 'machine'or a given amount of time. In the Poly/Metal1 deposition this time is ten minutes and in the Metal1/Metal2 deposition, it is 17 minutes. The DIV parameter controls the amount of grid added in the deposited layer.

One important stage of this process sequence is the reflow of the `Poly/Metal1` dielectric. This is used as a technique to remove sharp corners from the dielectric and improve the conformity of the `Metal1` deposition. The reflow properties for a material are determined by the viscosity (`visc.0`) and surface tension (`gamma.reflow`).

Since `Poly` and `Metal1` will run in the Z-direction they are both patterned by simple geometrical etches in ATHENA. One poly track is placed above an active area of the Silicon and one is placed on top of the field oxide. The Metal2 layer is unpatterned in ATHENA. It will extend in the X-direction. The limit of it's extent in the Z-direction is set in the next stage.

The final commands in ATHENA define each conductor to be an electrode. This electrode definition is carried through DEVEDIT3D into ATLAS. ATLAS will number the resultant capacitance matrix in the order of electrode definition in ATHENA. The backside of the silicon is not defined as an electrode. Doing this would provide a ground plane. It is not necessary to provide a ground plane in all 3D interconnect simulations.

The next stage of the example is to transfer the 2D data to DEVEDIT3D. DEVEDIT3D will extend the structure into 3D. The `Region` command in the DEVEDIT3D syntax modifies the extent of the Metal2 region. It is set to extend from Z=3 to Z=4 only. All other regions will extend the full depth of the structure. Z-planes at 0.5um intervals are defined by the `z.plane` command.

The DEVEDIT3D input syntax used here can be generated automatically from the graphical mode. It is highly recommended for beginners to use the graphical mode of DEVEDIT3D to make their edits to a 2D structure before advancing to this automated mode. Graphical DEVEDIT3D is called by the command DEVEDIT3D from the Unix prompt. Users can see the effect their changes have on the device before proceeding to ATLAS.

Once the 3D mesh has been created using DEVEDIT3D the structure is transferred to ATLAS. ATLAS will recognize the structure as 3D on reading it. There are no special commands required to switch ATLAS from 2D to 3D modes. To extract the full capacitance matrix between all six conductors the command, `models ex.cap`, is all that is needed.

The output of the ATLAS run will consist of two listings of the capacitance between each pair of electrodes. One is as English text and the other as a SPICE netlist. The final 3D structure is also saved to a solution file. This file can be viewed in 3D using TONYPLOT3D.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

20.1.6 interconnectex06.in: Multi-Level Metal Capacitances from Layout

Requires: MASKVIEWS/INTERCONNECT3D

This example demonstrates the extraction of parasitic capacitances between conductors in a multi-level interconnect structure. The structure is created using mask information from a layout file.

The structure used in this example is a multi-level interconnect structure. There are two Metal1 tracks running in the XZ plane and two metal1 tracks running in the XY plane. In order for the tracks to cross the tracks in the XZ plane are connected upwards to a Metal2 layer to make a bridge over the tracks in the XY plane. The 'bridge'consists of the Metal1/Metal2 via a short track of metal also in the XZ plane, and another metal1/metal2 via. All conductors are embedded in a block of silicon dioxide.

To see or edit the layout file, go to the DECKBUILD menu, **Tools->MaskViews->StartMaskViews**. Then load the file, **interconnectex06.lay**. From this layout, the cross-section file for this example was specified and saved. To create a section file, the user should push the **Write file** button in MASKVIEWS and select two corners of the ATLAS simulation area. In this example the section file was created earlier for the area from(0.,0.) to (10.,2.8) corners. The name of this file is specified with ATLAS syntax by the `layout` parameter of the `mesh` statement. The mesh in the X-Z plane will be created using layout file information and the values of NX and NZ, which also are specified in the mesh statement.

To build-up the structure in the Y-direction, the statements `addlayer` are used. For each layer to be included in the 3D interconnect structure, the parameter, `thick` for the layer thickness and `division` for the number of grid divisions of the layer in y-direction are specified. The complete 3D layer structure is built up in this manner.

When using the MASKVIEWS interface, it should be noted that **the user does not need to specify electrodes at all.** The program will find them automatically. However if users wish to override this fea-

ture they can specify ELECTRODES manually. To do this, the parameters X.PNT, Y.PNT, and Z.PNT are chosen (for parameter Y.PNT a default value of Y.PNT=0 is used). These parameters are the coordinates of a point which is inside the electrode. All the connected points of the interconnect structure will refer to the same electrode.

In the models statement the parameter, ex.cap, is all that is needed to extract the capacitance matrix between all electrodes. The output of ATLAS will be the listings of capacitance between each pair of electrodes and a SPICE sub-circuit in netlist format of the structure. The final 3D structure is saved to a solution file. This file can be viewed in 3D using TONYPLOT3D.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

20.1.7 interconnectex07.in: Via and Ground Planes from Layout

Requires: MASKVIEWS/INTERCONNECT3D

This example uses the MASKVIEWS layout interface to ATLAS to create a structure that has a via which runs through two ground planes and is sandwiched between two others.

The structure creation uses four mask layers and applies them as needed to create the materials sandwich.

Although the structure is different, all the details of this simulation are the same as described for the previous example. The layout file and cross section can be viewed and edited in the same manner. The layer structure is built up in ATLAS using addlayer. The command, model ex.cap, is used to enable the 3D capacitance extraction and the results appear as a listing of inter-metal capacitances, a SPICE sub-circuit netlist and a 3D structure file.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

20.1.8 interconnectex08.in: Resistance Calculation

Requires: INTERCONNECT3D

This example uses ATLAS command language to create a structure that is used for resistance analysis.

The command, models ex.res, is used to select that resistance calculations are to be done.

To load and run this example, select the **Load example** button while this text is displayed. The input file and an output file will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

20.1.9 interconnectex09.in: Capacitance Calculation from GDS-II Layout

Requires: MASKVIEWS/INTERCONNECT3D

This example uses ATLAS command language and a layout that was derived from a subset of a complete GDS-II layout to create a structure that is used for capacitance analysis.

The layout can be viewed by starting MASKVIEWS from the DECKBUILD **Tools** menu. Load the file, **interconnectex09.lay,** into MASKVIEWS to view the layout structure.

To load and run this example, select the **Load example** button while this text is displayed. The input file and an output file will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

20.1.10 interconnectex10.in: Effect of CD Variation and Misalignment

Requires: MASKVIEWS/INTERCONNECT3D

This example includes three separate runs of ATLAS. Each uses the same layout input file from MASKVIEWS. The structure is a simple quasi-2D dual metal structure. The first run is the nominal case. The second run uses the parameter, `delta.cd.x`, to adjust the dimensions of the M1 layer by 0.9x the original size. The third run uses the parameter, `misalign.x`, to misalign the M1 layer with respect to the M2 layer. The effect of these changes in dimension can be seen in the extracted capacitance values.

To load and run this example, select the **Load example** button while this text is displayed. The input file and an output file will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

This page is intentionally left blank

21.1. THERMAL: Thermal Distribution Application Examples

21.1.1 therm01.in: Simple Two Transistor Simulation

Requires: DEVEDIT3D/THERMAL3D

This example runs thermal simulation of a 2 transistors. The main sequential tasks are:

- specification of two neighboring transistor regions in DEVEDIT3D
- specification of heat sinks in in DEVEDIT3D
- interface to ATLAS
- selection of thermal models and material parameters
- solution of temperature distribution

The structure in this example is defined using DEVEDIT3D. Initially the structure was defined using the graphical mode of DevEdit3D. After the mesh was created a command file was saved from DEVEDIT3D enabling the structure to be re-created in DECKBUILD.

The structure used here consists of two neighboring transistors embedded in a substrate material. For thermal modeling the transistors are considered only as heat sources. No internal details of the transistors are considered. An Aluminum region is defined as an `electrode` on the bottom of the substrate material. This will act in ATLAS as a thermal rather than an electrical boundary condition.

The `go atlas` statement automatically interfaces DEVEDIT3D to ATLAS. On reading the three dimensional mesh file, ATLAS will automatically enter 3D mode.

The material statement is used to define the thermal conductivities of each region using the `tc.const` parameter. This parameter sets a constant value for thermal conductivity. Later examples show regions with temperature dependent thermal conductivities. The `power` parameter sets the thermal output of the region. This naturally defines these regions at heat sources.

The command, `models thermal,` is all that is required to enable the 3D thermal calculations. The final stage sets `solve t1=300`. This defines the temperature of the heat sink (or electrode) number 1. The final 3D thermal distribution is saved to the file specified by the `outfile` parameter. This can be viewed in TONYPLOT3D.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

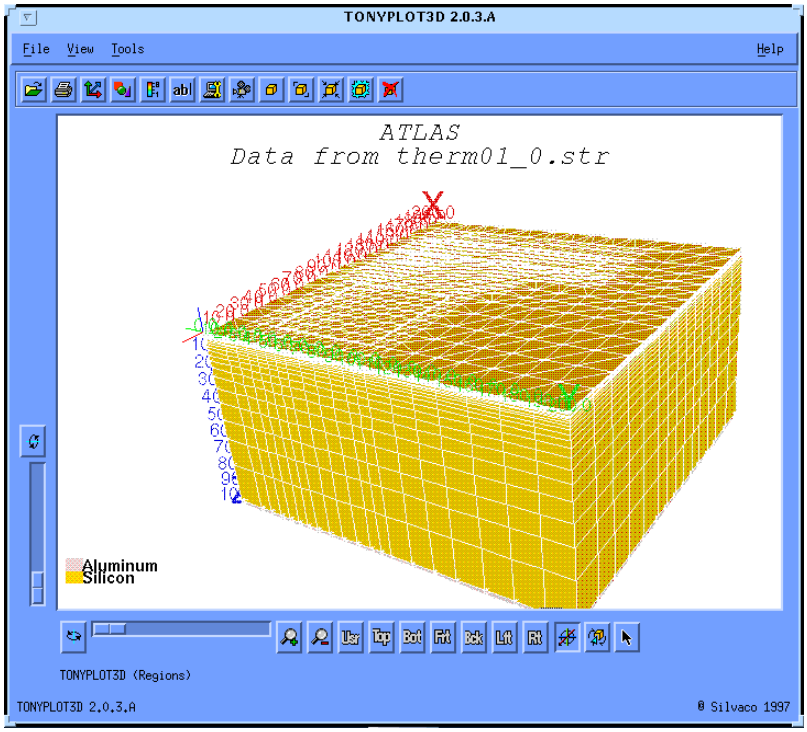


Figure 21.1: 3D Mesh and Geometry of a 3D substrate with two transistor power sources

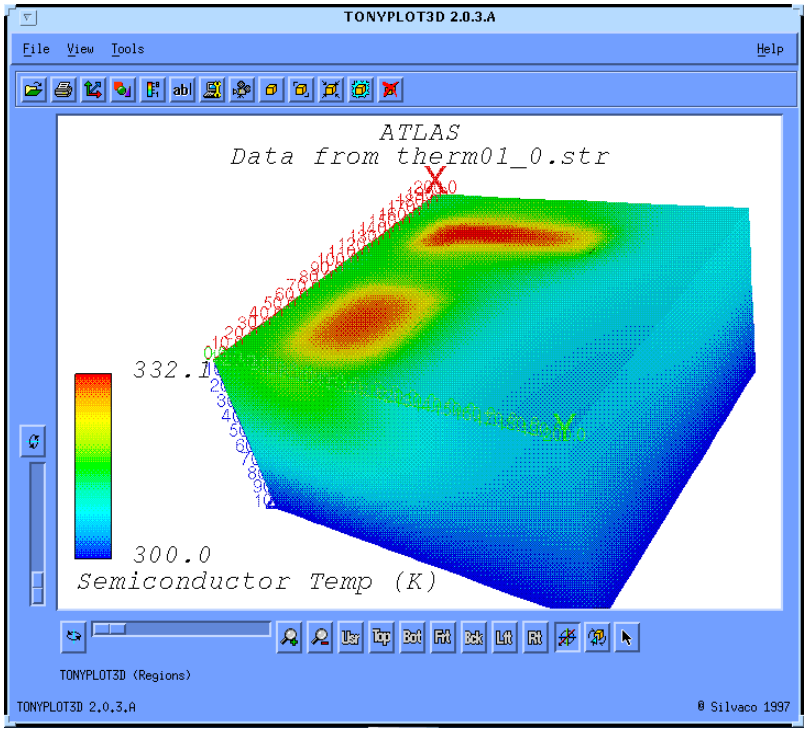


Figure 21.2: Contours of thermal distribution from the two power sources

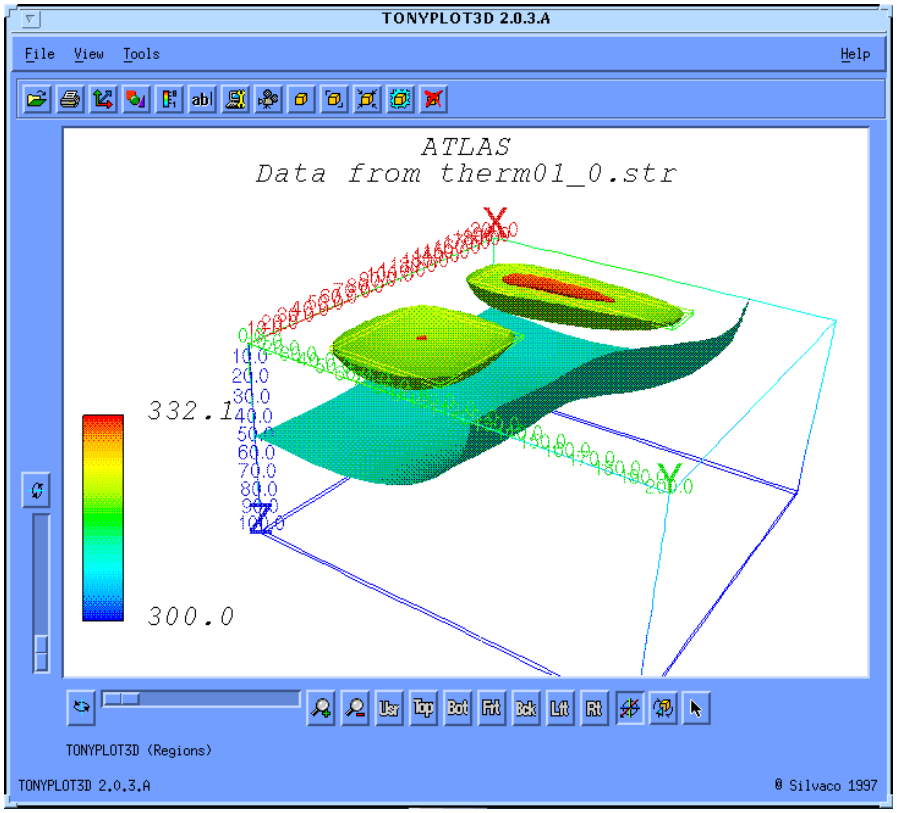


Figure 21.3: 3D iso-surfaces of temperature distribution in the substrate

Input File thermal/therm01.in:

```
1   go DevEdit
2
3   #
4   # ATLAS 3D Thermal simulation
5   # Two power transistor simulation
6   # SILVACO International April 1994
7   #
8
9   DevEdit version="2.0" library="1.14"
10
11  work.area left=0 top=0 right=200 bottom=200
12
13  # SILVACO Library V1.14
14
15  region reg=1 mat=Silicon color=0xffc000 pattern=0x4 Z1=0 Z2=100 \
16  points="0,0 200,0 200,200 0,200 0,0"
17  #
18  constr.mesh region=1 default
19
```

```
region reg=2 mat=Silicon color=0xffc000 pattern=0x4 Z1=0 Z2=2 \
points="20,30 85,30 85,90 20,90 20,30"
#
constr.mesh region=2 default

region reg=3 mat=Silicon color=0xffc000 pattern=0x4 Z1=0 Z2=2 \
points="130,30 165,30 165,140 140,140 140,55 130,55 130,30"
#
constr.mesh region=3 default

region reg=4 mat=Aluminum elec.id=1 color=0xffc0c0 pattern=0x7 Z1=100
  Z2=102 \
points="0,0 200,0 200,200 0,200 0,0"
#
constr.mesh region=4 default




# Set Meshing Parameters
#
base.mesh height=1000000 width=1000000
#
bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=0.001
  line.straightening=1 align.points when=automatic
#
imp.refine min.spacing=0.02 z=0
#
constr.mesh max.angle=90 max.ratio=300 max.height=20 \
max.width=20 min.height=0.0001 min.width=0.0001
#
constr.mesh type=Semiconductor default
#
constr.mesh type=Insulator default
#
constr.mesh type=Metal default
#
constr.mesh type=Other default
#
constr.mesh region=1 default
#
constr.mesh region=2 default
#
constr.mesh region=3 default
#
```

```
62  constr.mesh region=4 default
63  #
64  # Perform mesh operations
65  #
66  Mesh Mode=MeshBuild
67  refine mode=both x1=8.04953560371517 y1=22.9102167182663
      x2=91.0216718266254 y2=96.594427244582
68  refine mode=both x1=123.529411764706 y1=22.6006191950464
      x2=174.922600619195 y2=145.510835913313
69  refine mode=both x1=99.6904024767802 y1=21.671826625387
      x2=113.931888544892 y2=107.430340557276
70
71  imp.refine min.spacing=0.02 z=0
72
73  constr.mesh max.angle=90 max.ratio=300 max.height=20 \
74  max.width=20 min.height=0.0001 min.width=0.0001
75  #
76  constr.mesh type=Semiconductor default
77  #
78  constr.mesh type=Insulator default
79  #
80  constr.mesh type=Metal default
81  #
82  constr.mesh type=Other default
83
84  z.plane z=0 spacing=0.5
85  #
86  z.plane z=2 spacing=0.5
87  #
88  z.plane z=50 spacing=10
89  #
90  z.plane z=100 spacing=20
91  #
92  z.plane max.spacing=1000000 max.ratio=10
93
94  base.mesh height=1000000 width=1000000
95
96  bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=0.001
      line.straightening=1 align.Points when=automatic
97
98
99  go atlas
100 #
101 material region=1 tc.const=1.2
```

```
material region=2 tc.const=1. power=0.35
material region=3 tc.const=1.1 power =0.4
#
models thermal
method
#
solve t1=300 outf=therm01_0.str
#
quit
```

21.1.2 therm02.in: Eight Power Source Structure

Requires: DEVEDIT3D/THERMAL3D

This example runs thermal simulation of a 2 layer structure with 8 power sources. The main sequential tasks are:

- specification of all ten regions in DEVEDIT3D
- specification of heat sink in in DEVEDIT3D
- interface to ATLAS
- selection of thermal models and material parameters
- solving of temperature distribution

This example is similar to the device described in **Thermal Simulation and Design of a GaAs HBT Sample and Hold Circuit** by Ken Poulton et al. The device technology is from the paper by R. Nubling in Technical Digest of 1990 GaAs IC Symposium, pp 53-56.

The structure in this example is defined using DEVEDIT3D in a similar manner to the previous example. Initially the structure was defined using the graphical mode of DEVEDIT3D. After the mesh was created a command file was saved from DEVEDIT3D enabling the structure to be re-created in DECKBUILD.

Once in ATLAS, the `models thermal` statement is used to indicate that this is a heat distribution simulation. The `material` statement is used to set the heat power output of each of the eight heat producing regions in this structure. Region #10 is defined to have a constant thermal conductivity of 1.0. The `solve t1=300` statement sets the thermal boundary condition on the electrode to be 300K and then solves the heat flow equation self-consistently in the structure. The resulting 3D heat distribution can be shown in TONYPLOT3D.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

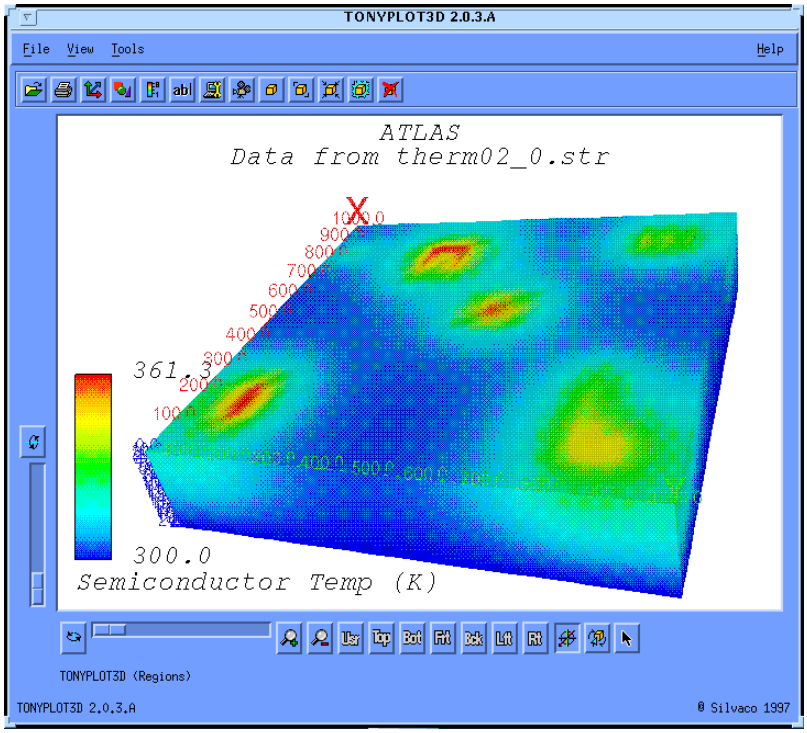


Figure 21.4: Contours of thermal distribution in a 3D substrate containing eight power sources

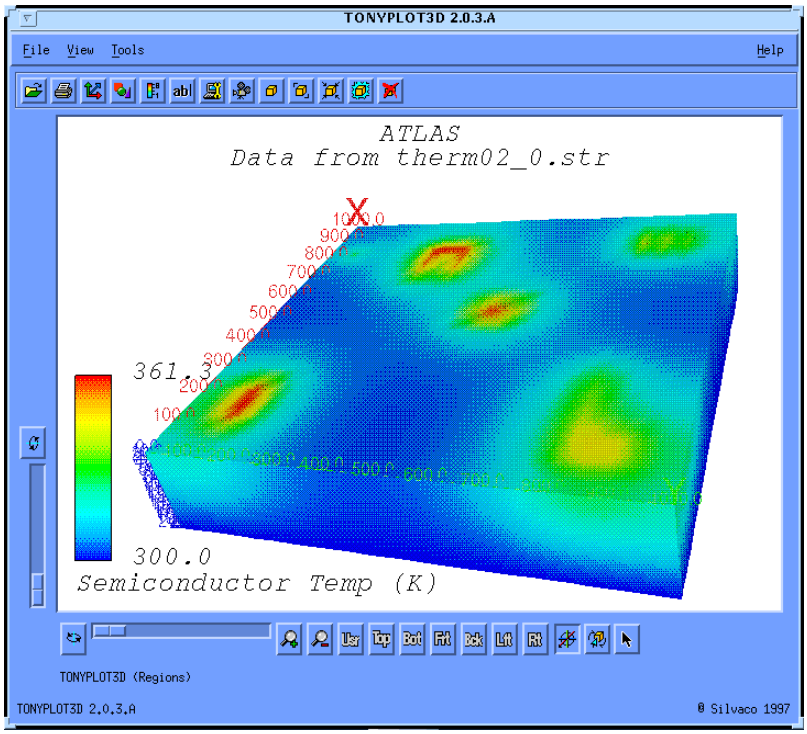


Figure 21.5: 3D iso-surfaces of temperature distribution in the substrate

Input File thermal/therm02.in:

```
1    go DevEdit
```

```
2
3   #
4   # ATLAS 3D Thermal simulation
5   # 2 layer structure with 8 elements dissipating energy
6   # Silvaco International April 1994
7   #
8
9   DevEdit version="2.0" library="1.14"
10
11  work.area left=0 top=0 right=1000 bottom=1000
12
13  # SILVACO Library V1.14
14
15  region reg=1 mat=Silicon color=0xffc000 pattern=0x4 Z1=0 Z2=10 \
16  points="0,0 1000,0 1000,1000 0,1000 0,0"
17  #
18  constr.mesh region=1 default
19
20  region reg=2 mat=Silicon color=0xffc000 pattern=0x4 Z1=0 Z2=2 \
21  points="500,500 600,500 600,600 500,600 500,500"
22  #
23  constr.mesh region=2 default
24
25  region reg=3 mat=Silicon color=0xffc000 pattern=0x4 Z1=0 Z2=2 \
26  points="700,280 870,280 870,400 700,400 700,370 830,370 830,310 700,310
      700,280"
27  #
28  constr.mesh region=3 default
29
30  region reg=4 mat=Silicon color=0xffc000 pattern=0x4 Z1=0 Z2=2 \
31  points="100,90 260,90 260,160 100,160 100,90"
32  #
33  constr.mesh region=4 default
34
35  region reg=5 mat=Silicon color=0xffc000 pattern=0x4 Z1=0 Z2=2 \
36  points="60,790 350,790 350,850 170,850 170,930 60,930 60,790"
37  #
38  constr.mesh region=5 default
39
40  region reg=6 mat=Silicon color=0xffc000 pattern=0x4 Z1=0 Z2=2 \
41  points="800,50 860,50 860,90 800,90 800,50"
42  #
43  constr.mesh region=6 default
```

```
44
45 region reg=7 mat=Silicon color=0xffc000 pattern=0x4 Z1=0 Z2=2 \
46 points="800,850 920,850 920,870 800,870 800,850"
47 #
48 constr.mesh region=7 default
49
50 region reg=8 mat=Silicon color=0xffc000 pattern=0x4 Z1=0 Z2=2 \
51 points="800,800 920,800 920,820 800,820 800,800"
52 #
53 constr.mesh region=8 default
54
55 region reg=9 mat=Silicon color=0xffc000 pattern=0x4 Z1=0 Z2=2 \
56 points="800,900 920,900 920,920 800,920 800,900"
57 #
58 constr.mesh region=9 default
59
60 region reg=10 mat="Silicon Oxide" color=0xff pattern=0x2 Z1=10 Z2=200 \
61 points="0,0 1000,0 1000,1000 0,1000 0,0"
62 #
63 constr.mesh region=10 default
64
65 region reg=11 name=substrate mat=Aluminum elec.id=1 work.func=0 col-
     or=0xffc8c8 pattern=0x7 Z1=200 Z2=201 \
66 points="0,0 1000,0 1000,1000 0,1000 0,0"
67 #
68 constr.mesh region=11 default
69
70 # Set Meshing Parameters
71 #
72 base.mesh height=1000000 width=1000000
73 #
74 bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=0.001
     line.straightening=1 align.points when=automatic
75 #
76 imp.refine min.spacing=0.02 z=0
77 #
78 constr.mesh max.angle=90 max.ratio=300 max.height=100 \
79 max.width=100 min.height=0.0001 min.width=0.0001
80 #
81 constr.mesh type=Semiconductor default
82 #
83 constr.mesh type=Insulator default
84 #
85 constr.mesh type=Metal default
```

```
#
constr.mesh type=Other default
#
constr.mesh region=1 default
#
constr.mesh region=2 default
#
constr.mesh region=3 default
#
constr.mesh region=4 default
#
constr.mesh region=5 default
#
constr.mesh region=6 default
#
constr.mesh region=7 default
#
constr.mesh region=8 default
#
constr.mesh region=9 default
#
constr.mesh region=10 default
#
constr.mesh region=11 default
#
# Perform mesh operations
#
Mesh Mode=MeshBuild
refine mode=both x1=26.3157894736842 y1=29.4117647058824
  x2=315.789473684211 y2=246.130030959752
refine mode=both x1=755.417956656347 y1=9.28792569659443
  x2=896.284829721362 y2=171.826625386997
refine mode=both x1=668.730650154799 y1=210.526315789474
  x2=891.640866873065 y2=473.684210526316
refine mode=both x1=450.46439628483 y1=430.340557275542
  x2=679.566563467492 y2=613.003095975232
refine mode=both x1=766.25386996904 y1=721.362229102167
  x2=961.300309597523 y2=975.232198142415
refine mode=both x1=29.4117647058824 y1=726.006191950464
  x2=388.544891640867 y2=973.684210526316

imp.refine min.spacing=0.02 z=0

constr.mesh max.angle=90 max.ratio=300 max.height=100 \
max.width=100 min.height=0.0001 min.width=0.0001
```

```
#
constr.mesh type=Semiconductor default
#
constr.mesh type=Insulator default
#
constr.mesh type=Metal default
#
constr.mesh type=Other default

z.plane z=2 spacing=1
#
z.plane z=10 spacing=2.5
#
z.plane z=200 spacing=40
#
z.plane max.spacing=1000000 max.ratio=100

base.mesh height=1000000 width=1000000

bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=0.001
    line.straightening=1 align.Points when=automatic


go atlas

models thermal
method

material region=1
material region=2 power=1.3
material region=3 power=2.0
material region=4 power=1.5
material region=5 power=1.8
material region=6 power=0.2
material region=7 power=0.3
material region=8 power=0.4
material region=9 power=0.3
material region=10 tc.const=1

solve t1=300 outf=therm02_0.str

quit

```

21.1.3 therm03.in: Layers with Temperature Dependent Conductivity

Requires: DEVEDIT3D/THERMAL3D

This example runs thermal simulation of a two layer structure with eight power sources where the thermal conductivity of region #1(the first layer) and region #10(the second layer) is a function of temperature.

The structure and syntax used in this example is exactly the same as that used in the previous example with the following exception. The thermal conductivity of region #1 is defined to be dependent on temperature by the statement, `tcon.polynom`. This sets a polynomial dependence of thermal conductivity as described in the manual.

In region #10, the thermal conductivity is set according to a simpler power law. The parameter, `tcon.power`, is used to indicate this. Parameters of the power law are set using the `tc.c0` and `tc.npow`. Details of the equations used can be found in the models description of THERMAL3D.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

21.1.4 therm04.in: GaAs IC Structure

Requires: DEVEDIT3D/THERMAL3D

This example runs thermal simulation of a GaAs IC structure with three heat producing devices.

The device consists of three GaAs regions defined as heat sources in a GaAs substrate. A single thermal contact is applied to the bottom of the substrate region. The structure is prepared using DevEdit3D in a similar manner to the previous examples in this section.

In ATLAS the `material` statement is used to define the power output of two of the three GaAs regions. The substrate is defined as having a temperature dependent thermal conductivity according to the `tcon.power` parameter.

This example is designed to show the effect of the heating produced by neighboring devices on a non-heat producing transistor. The third GaAs region (defined as region #4) has no heat power output. However, the solution file will show that the device is heated by conduction from the neighboring devices.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

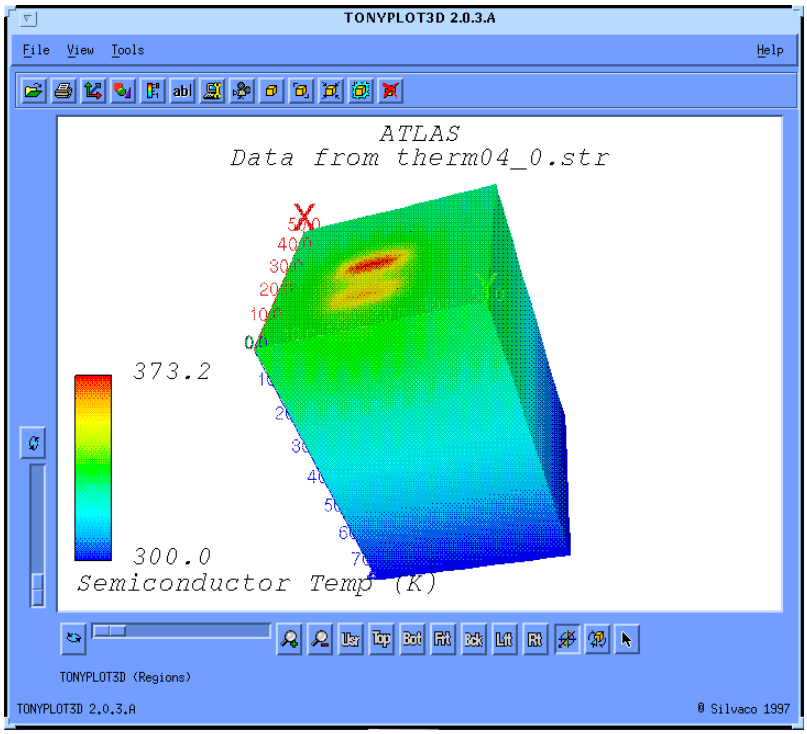


Figure 21.6: Thermal distribution in a GaAs substrate from a set of three power sources on the surface

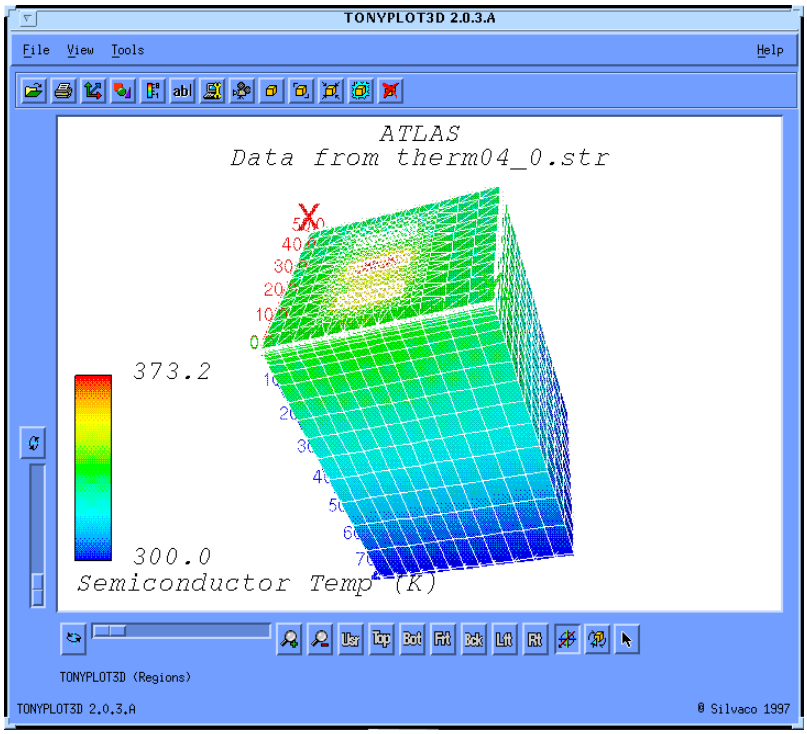


Figure 21.7: 3D GaAs substrate structure with contours of temperature and mesh

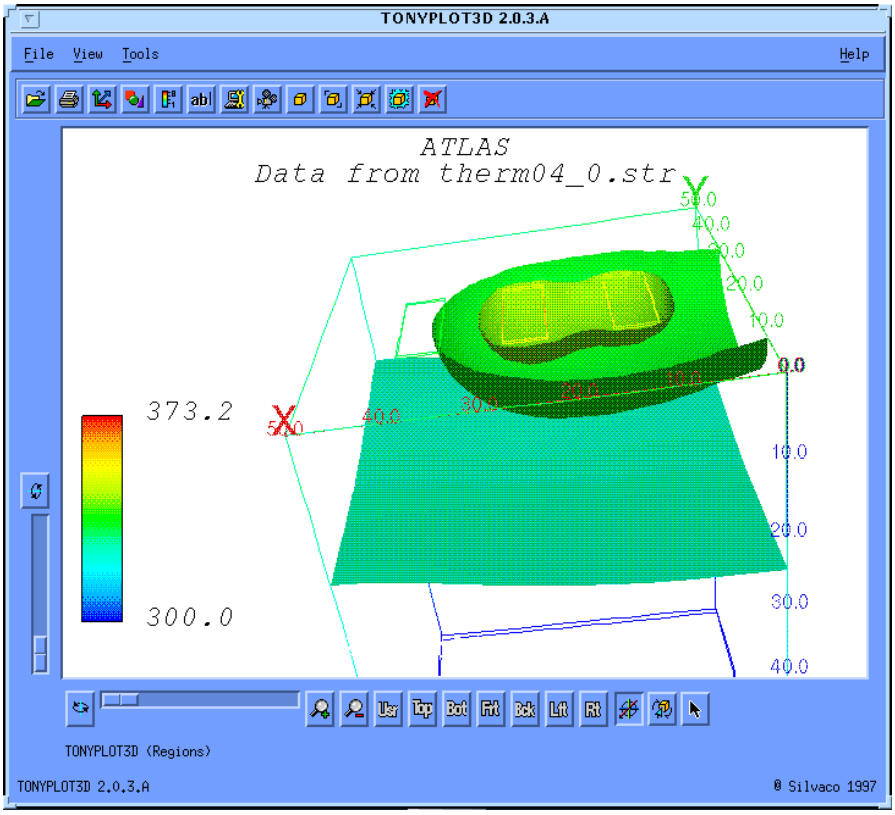


Figure 21.8: 3D iso-surfaces of temperature distribution from the three sources in a GaAs substrate

Input File thermal/therm04.in:

```
1   go DevEdit
2
3   DevEdit version="2.1" library="1.15"
4
5   work.area left=0 top=0 right=50 bottom=50
6
7   #  DevEdit V2.1 L1.15
8
9   region reg=1 mat=GaAs color=0xffcb pattern=0x9 Z1=0 Z2=75 \
10  points="0,50 0,0 50,0 50,50 0,50"
11  #
12  constr.mesh region=1 default
13
14  region reg=2 mat=GaAs color=0xffcb pattern=0x9 Z1=0 Z2=0.3 \
15  points="10,15 15,15 15,30 10,30 10,15"
16  #
17  constr.mesh region=2 default
18
19  region reg=3 mat=GaAs color=0xffcb pattern=0x9 Z1=0 Z2=0.3 \
```

```
20 points="22.5,15 27.5,15 27.5,30 22.5,30 22.5,15"
21 #
22 constr.mesh region=3 default
23
24 region reg=4 name=GaAs mat=GaAs color=0xffcb pattern=0x9 Z1=0 Z2=0.3 \
25 points="40,15 40,30 35,30 35,15 40,15"
26 #
27 constr.mesh region=4 default
28
29 region reg=5 name=substrate mat=Silver elec.id=1 work.func=0 color=0x7fff
     pattern=0xc Z1=75 Z2=76 \
30 points="0,0 50,0 50,50 0,50 0,0"
31 #
32 constr.mesh region=5 default
33
34
35
36 # Set Meshing Parameters
37 #
38 base.mesh height=1000000 width=1000000
39 #
40 bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=0.001
     line.straightening=1 align.points when=automatic
41 #
42 imp.refine min.spacing=0.02 z=0
43 #
44 constr.mesh max.angle=90 max.ratio=300 max.height=6 \
45 max.width=6 min.height=0.0001 min.width=0.0001
46 #
47 constr.mesh type=Semiconductor default
48 #
49 constr.mesh type=Insulator default
50 #
51 constr.mesh type=Metal default
52 #
53 constr.mesh type=Other default
54 #
55 constr.mesh region=1 default
56 #
57 constr.mesh region=2 default
58 #
59 constr.mesh region=3 default
60 #
61 constr.mesh region=4 default
```

```
#
constr.mesh region=5 default
#
# Perform mesh operations
#
Mesh Mode=TensorProduct
refine mode=both x1=6.3 y1=12.66 x2=43.13 y2=32.93

imp.refine min.spacing=0.02 z=0

constr.mesh max.angle=90 max.ratio=300 max.height=6 \
max.width=6 min.height=0.0001 min.width=0.0001
#
constr.mesh type=Semiconductor default
#
constr.mesh type=Insulator default
#
constr.mesh type=Metal default
#
constr.mesh type=Other default

z.plane z=0.3 spacing=0.15
#
z.plane z=15 spacing=5
#
z.plane z=75 spacing=10
#
z.plane max.spacing=1000000 max.ratio=100

base.mesh height=1000000 width=1000000

bound.cond !apply max.slope=28 max.ratio=300 rnd.unit=0.001
  line.straightening=1 align.Points when=automatic

go atlas

models thermal

material reg=1 tcon.power
material reg=2 power=0.02
material reg=3 power=0.03

method newton
```

```
104
105 solve t1=300 outf=therm04_0.str
106
107 quit
```

21.1.5 therm05.in: Simple Simulation with Temperature Stepping

Requires: DEVEDIT3D/THERMAL3D

This example runs thermal simulation of a 2 transistors. The main sequential tasks are:

- specification of two neighboring transistor regions in DEVEDIT3D
- specification of heat sinks in in DEVEDIT3D
- interface to ATLAS
- selection of thermal models and material parameters
- specification of temperature stepping parameters
- solution of temperature distribution

The structure in this example is defined using DEVEDIT3D. Initially the structure was defined using the graphical mode of DEVEDIT3D. After the mesh was created a command file was saved from DEVEDIT3D enabling the structure to be re-created in DECKBUILD.

The structure used here consists of two neighboring transistors embedded in a substrate material. For thermal modeling the transistors are considered only as heat sources. No internal details of the transistors are considered. An Aluminum region is defined as an electrode on the bottom of the substrate material. This will act in ATLAS as a thermal rather than an electrical boundary condition.

The `go atlas` statement automatically interfaces DEVEDIT3D to ATLAS. On reading the three dimensional mesh file, ATLAS will automatically enter 3D mode.

The material statement is used to define the thermal conductivities of each region using the `tc.const` parameter. This parameter sets a constant value for thermal conductivity. Later examples show regions with temperature dependent thermal conductivities. The `power` parameter sets the thermal output of the region. This naturally defines these regions at heat sources.

The command, `models thermal,` is all that is required to enable the 3D thermal calculations. The final stage sets `solve t1=300`. This defines the temperature of the heat sink (or electrode) number 1. In this example the temperature on electrode one is stepped by adding the `TEMPSTEP NSTEPS` and `electrode` parameters. `TEMPSTEP` determines the size of the temperature step and the `NSTEPS` gives the number of temperature steps to be performed. The `electrode` parameter identifies which electrode to step as more than one electrode temperature can be defined on the `solve` statement. The 3D thermal distributions are saved to the file specified by the `outfile` parameter. The output file name rightmost characters will be incremented alphanumerically so a different file will be saved for each step. These can be viewed in TONYPLOT3D.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

21.1.6 therm06.in: Simple Simulation with Power Stepping

Requires: DEVEDIT3D/THERMAL3D

This example runs thermal simulation of two transistors. The main sequential tasks are:

- specification of two neighboring transistor regions in DEVEDIT3D

- specification of heat sinks in in DEVEDIT3D
- interface to ATLAS
- selection of thermal models and material parameters
- specification of the power stepping parameters
- solution of temperature distribution at different thermal powers

The structure in this example is defined using DEVEDIT3D. Initially the structure was defined using the graphical mode of DEVEDIT3D. After the mesh was created, a command file was saved from DEVEDIT3D enabling the structure to be re-created in DECKBUILD.

The structure used here consists of two neighboring transistors embedded in a substrate material. For thermal modeling the transistors are considered only as heat sources. No internal details of the transistors are considered. An Aluminum region is defined as an `electrode` on the bottom of the substrate material. This will act in ATLAS as a thermal rather than an electrical boundary condition.

The `go atlas` statement automatically interfaces DEVEDIT3D to ATLAS. On reading the three dimensional mesh file, ATLAS will automatically enter 3D mode.

The material statement is used to define the thermal conductivities of each region using the `tc.const` parameter. This parameter sets a constant value for thermal conductivity. Later examples show regions with temperature dependent thermal conductivities. The `power` parameter sets the thermal output of the region. This naturally defines these regions at heat sources. The region thermal power can also be defined in the `solve` statement for power stepping as shown below.

The command, `models thermal,` is all that is required to enable the 3D thermal calculations. The final stage sets `solve t1=300`. This defines the temperature of the heat sink (or electrode) number 1. The region thermal power can be specified by adding the `POWERx` parameter where **x** is the region number. This has the same effect as in the `material` statement and will override the value if already set. In this example the thermal power of region #2 is stepped by adding the `POWERSTEP NSTEPS` and `STEPREGION` parameters.

`POWER2` defines the initial thermal power for region #2. `POWERSTEP` determines the size of the power step, `NSTEPS` sets the number of power steps to be performed and the `STEPREGION` option determines which region to step as the thermal power of any region may be specified on the `solve` statement.

The final 3D thermal distribution is saved to the file specified by the `outfile` parameter. The output file name rightmost characters will be incremented alphanumerically so a different file will be saved for each step. These can be viewed in TONYPLOT3D.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

21.1.7 therm07.in: Simple Simulation with Temperature and Power Stepping

Requires: DEVEDIT3D/THERMAL3D

This example runs thermal simulation of a 2 transistors. The main sequential tasks are:

- specification of two neighboring transistor regions in DEVEDIT3D
- specification of heat sinks in in DEVEDIT3D
- interface to ATLAS
- selection of thermal models and material parameters

- specification of temperature and power stepping parameters
- solution of temperature distributions at different temperatures and thermal powers

The structure in this example is defined using DEVEDIT3D. Initially the structure was defined using the graphical mode of DEVEDIT3D. After the mesh was created a command file was saved from DEVEDIT3D enabling the structure to be re-created in DECKBUILD.

The structure used here consists of two neighboring transistors embedded in a substrate material. For thermal modeling the transistors are considered only as heat sources. No internal details of the transistors are considered. An Aluminum region is defined as an `electrode` on the bottom of the substrate material. This will act in ATLAS as a thermal rather than an electrical boundary condition.

The `go atlas` statement automatically interfaces DEVEDIT3D to ATLAS. On reading the three dimensional mesh file, ATLAS will automatically enter 3D mode.

The material statement is used to define the thermal conductivities of each region using the `tc.const` parameter. This parameter sets a constant value for thermal conductivity. Later examples show regions with temperature dependent thermal conductivities. The `power` parameter sets the thermal output of the region. This naturally defines these regions at heat sources.

The command, `models thermal`, is all that is required to enable the 3D thermal calculations. The final stage sets `solve t1=300`. This defines the temperature of the heat sink (or electrode) number 1. In this example the temperature on electrode 1 and the power of region 2 are stepped by adding the `POWERSTEP`, `TEMPSTEP`, `NSTEPS`, `STEPREGION` and `electrode` parameters. POWERSTEP and `TEMPSTEP` determines the size of the power and temperature steps and the `NSTEPS` parameter gives the number of steps to be performed. The `STEPREGION` and `electrode` parameters identifies which region and electrodes to step as more than one region; thermal power and electrode temperature can be defined on the `solve` statement. The final 3D thermal distribution is saved to the file specified by the `outfile` parameter. The output file name rightmost characters will be incremented alphanumerically so a different file will be saved for each step. These can be viewed in TONYPLOT3D.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

This page is intentionally left blank

22.1. FERRO: Ferroelectric Application Examples

22.1.1 ferroex01.in: Material Parameter Effects in 1D MOS Capacitor

Requires: SPISCES/FERRO

This simulation demonstrates the effects of various ferroelectric material settings on the CV characterisitics of a Metal-Ferroelectric-Silicon Capacitor.

It demonstrates:

- Basic structure definition using ATLAS syntax
- Definition of PZT material parameters
- Application of ferroelectric model
- Using probe to analyse polarization
- Plotting output curves and structures.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory. Once loaded into DECKBUILD, select the **run** button to execute the example.

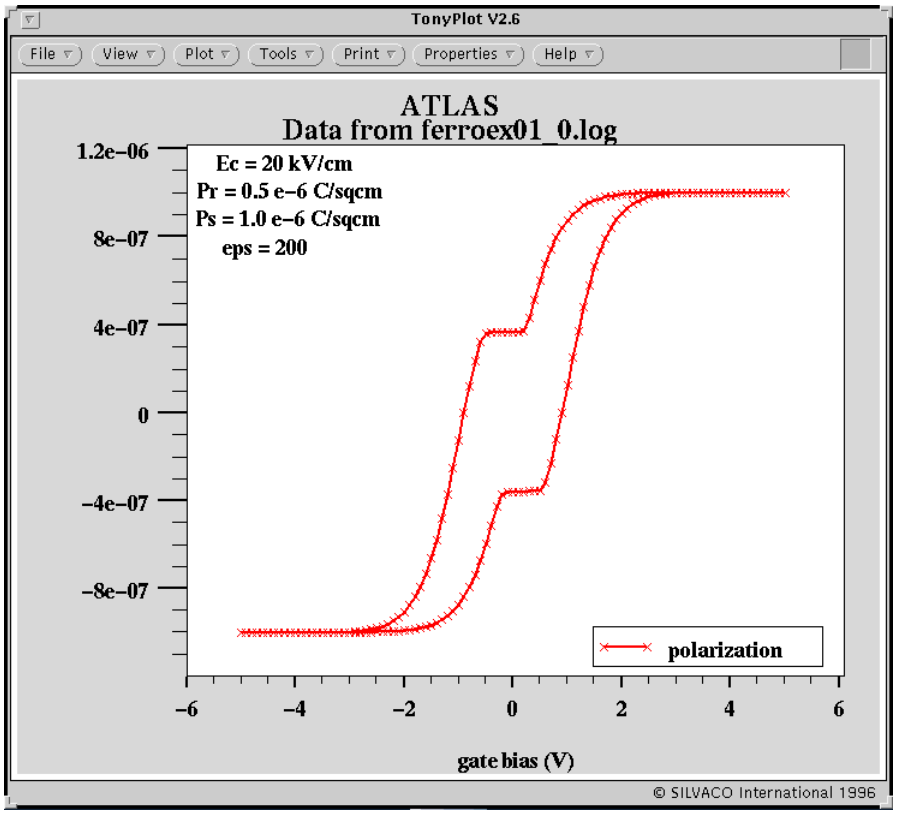


Figure 22.1: Polarization of the Ferroelectric film versus applied bias

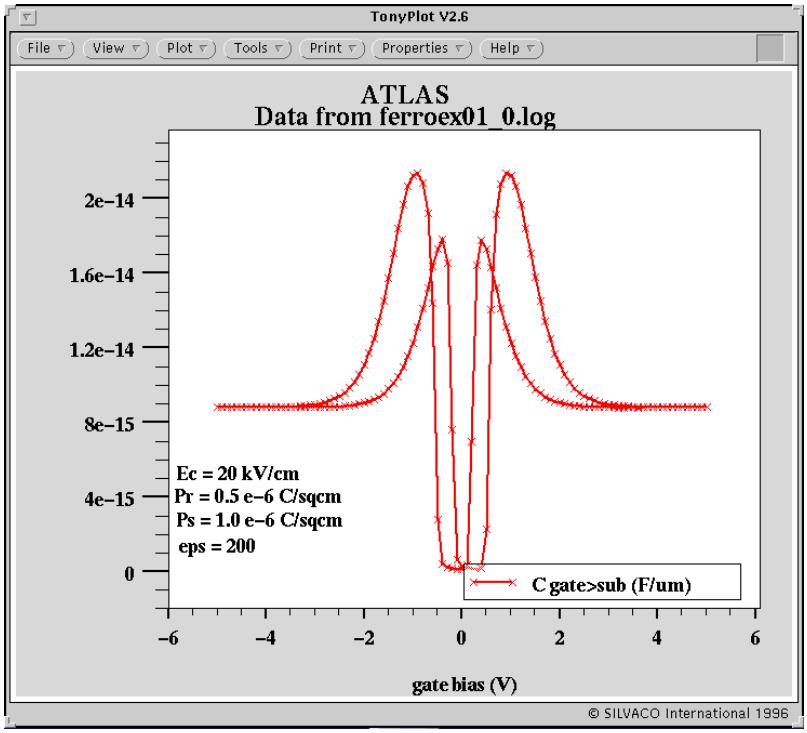


Figure 22.2: CV curve for a metal/ferroelectric/semiconductor capacitor

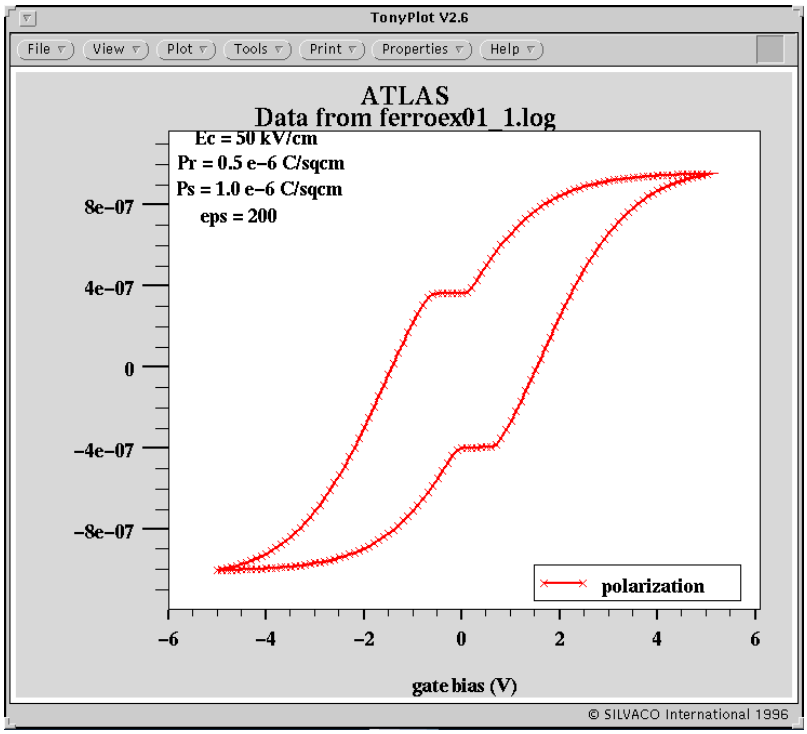


Figure 22.3: Effect of EC in the ferroelectric model on polarization. This plot should be compared to the first for this example

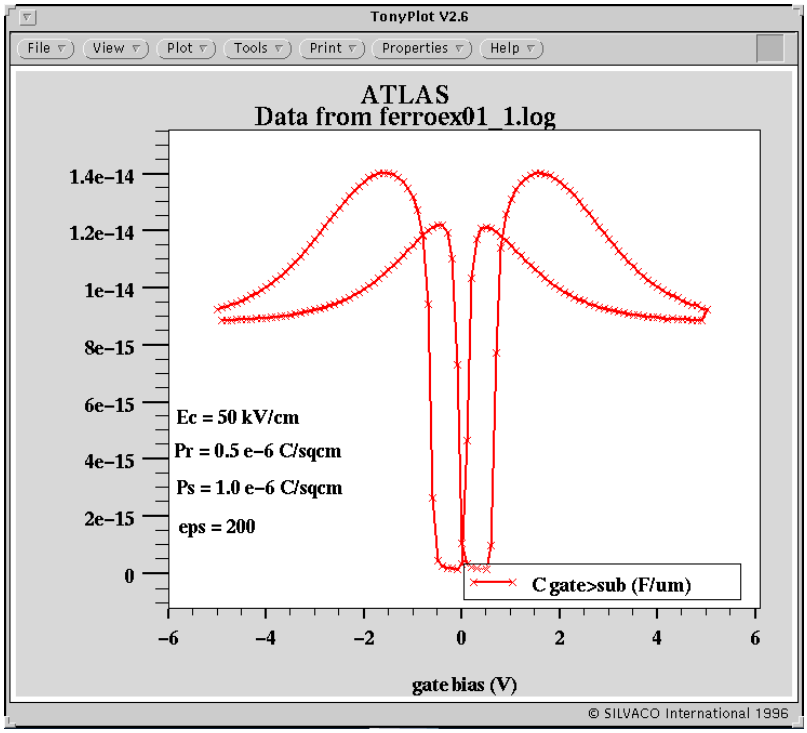


Figure 22.4: Effect of EC on capacitance. This plot should be compared to the second for this example

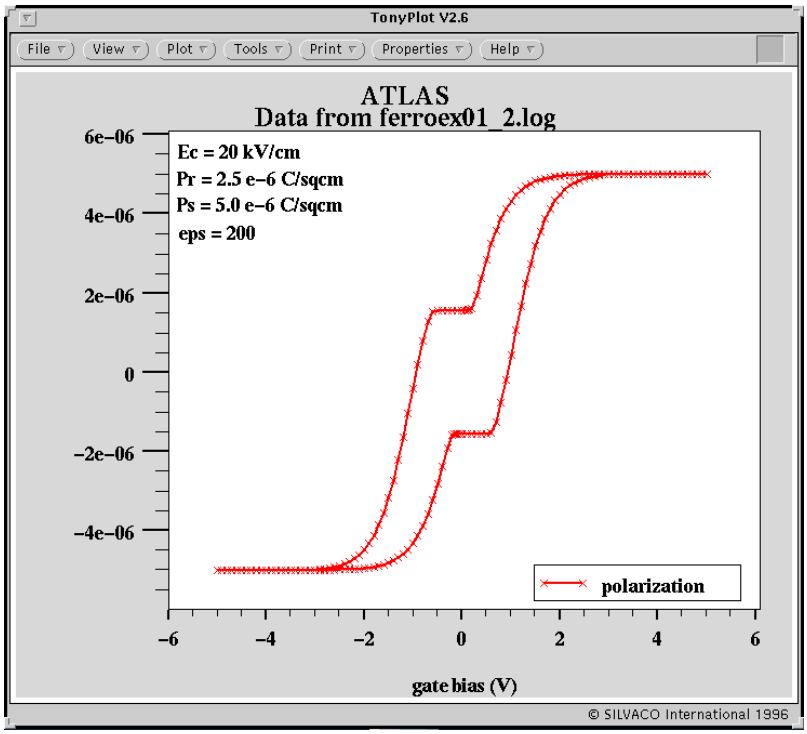


Figure 22.5: Effect of PR and PC in the ferroelectric model on polarization. This plot should be compared to the first for this example

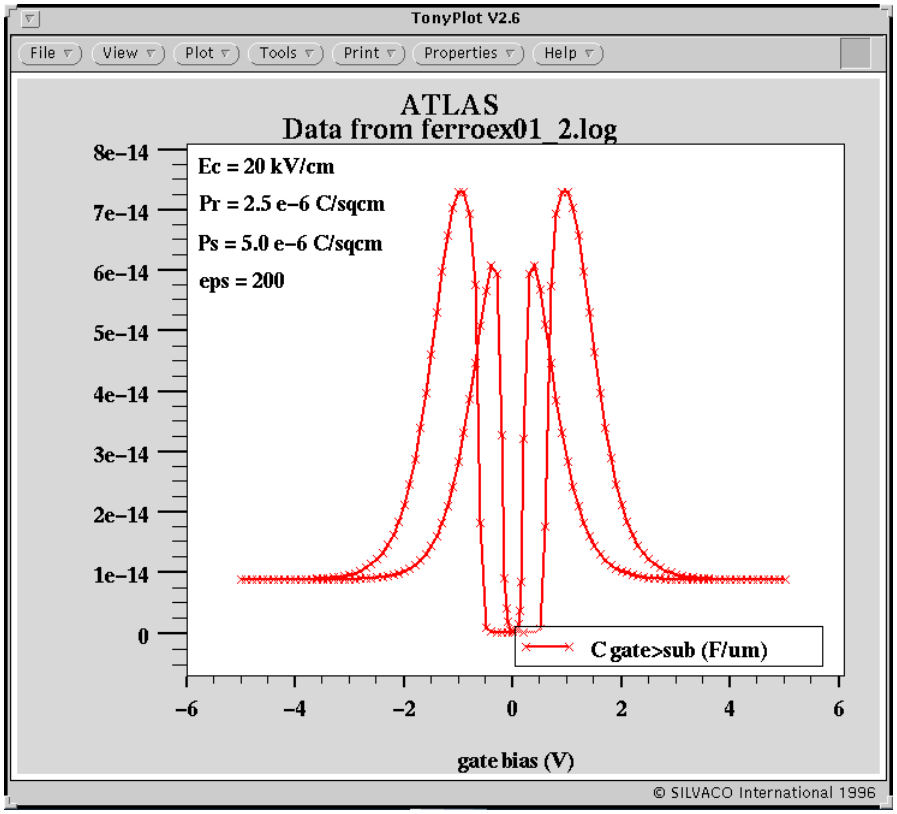


Figure 22.6: Effect of PR and PC on capacitance. This plot should be compared to the second for this example

Input File ferro/ferroex01.in :

```
1   go atlas
2   # Ferroelectric MOS Capacitor example (1D)
3   # Demonstrates effect of various material settings
4   #
5   mesh outf=ferroex01_0.str
6   #
7   x.m loc=0.0 spacing=1.0
8   x.m loc=1.0 spac=1.0
9   #
10  y.m loc=-0.2 spacing=0.2
11  y.m loc=0.0 spacing=0.2
12  y.m loc=5.0 spac=1.0
13  #
14  region num=1 silicon y.min=0.0
15  region num=2 oxide y.max=0.0
16  #
17  electrode top name=gate
18  electrode bottom name=sub
19  #
```

```
doping region=1 conc=1e15 uniform p.type
#
models region=2 ferro
material region=2 ferro.ec=20e3 ferro.pr=0.5e-6 ferro.ps=1.0e-6 fer-
  ro.epsf=200
#
method newton trap
#
solve vgate=-5
probe x=0 y=-0.1 dir=90 polar
#
log outf=ferroex01_0.log master
#
solve  name=gate vfinal=5 vstep=0.1 ac freq=1e-6 direct
material region=2 ferro.ec=-20e3
solve  name=gate vfinal=-5 vstep=-0.1 ac freq=1e-6 direct
#
tonyplot -st ferroex01_0.log -set ferroex01_0.set
tonyplot -st ferroex01_0.log -set ferroex01_1.set
#
go atlas
#
mesh outf=ferroex01_1.str
#
x.m loc=0.0 spacing=1.0
x.m loc=1.0 spac=1.0
#
y.m loc=-0.2 spacing=0.2
y.m loc=0.0 spacing=0.2
y.m loc=5.0 spac=1.0
#
region num=1 silicon y.min=0.0
region num=2 oxide y.max=0.0
#
electrode top name=gate
electrode bottom name=sub
#
doping region=1 conc=1e15 uniform p.type
#
models region=2 ferro
material region=2 ferro.ec=50e3 ferro.pr=0.5e-6 ferro.ps=1.0e-6 fer-
  ro.epsf=200
#
method newton trap
```

```
#
solve vgate=-5
#
probe x=0 y=-0.1 dir=90 polar
#
log outf=ferroex01_1.log master
#
solve  name=gate vfinal=5 vstep=0.1 ac freq=1e-6 direct
material region=2 ferro.ec=-50e3
solve  name=gate vfinal=-5 vstep=-0.1 ac freq=1e-6 direct
#
tonyplot  -st ferroex01_1.log  -set ferroex01_2.set
tonyplot  -st ferroex01_1.log  -set ferroex01_3.set
#
go atlas
#
mesh outf=ferroex01_2.str
#
x.m loc=0.0 spacing=1.0
x.m loc=1.0 spac=1.0
#
y.m loc=-0.2 spacing=0.2
y.m loc=0.0 spacing=0.2
y.m loc=5.0 spac=1.0
#
region num=1 silicon y.min=0.0
region num=2 oxide y.max=0.0
#
electrode top name=gate
electrode bottom name=sub
#
doping region=1 conc=1e15 uniform p.type
#
models region=2 ferro
material region=2 ferro.ec=20e3 ferro.pr=2.5e-6 ferro.ps=5.0e-6 fer-
  ro.epsf=200
#
method newton trap
#
solve vgate=-5
#
probe x=0 y=-0.1 dir=90 polar
#
```

```
104 log outf=ferroex01_2.log master
105 #
106 solve  name=gate vfinal=5 vstep=0.1 ac freq=1e-6 direct
107 material region=2 ferro.ec=-20e3
108 solve  name=gate vfinal=-5 vstep=-0.1 ac freq=1e-6 direct
109 save outf=ferroex01_2.str
110 #
111 tonyplot -st ferroex01_2.log -set ferroex01_4.set
112 tonyplot -st ferroex01_2.log -set ferroex01_5.set
113 #
114 quit
```

22.1.2 ferroex02.in: Threshold Shift in 2D FET

Requires: SPISCES/FERRO

This simulation demonstrates the threshold voltage shift due to ferroelectric polarization in the insulating gate dielectric of an FET.

- Basic structure definition using ATLAS syntax
- Definition of PZT material parameters
- Application of ferroelectric model
- Using probe to analyse polarisation
- Plotting output curves and structures.

To load and run this example, select the **Load example** button while this text is displayed. The input file and several support files will be copied to your current working directory at this time. Once loaded into DECKBUILD, select the **run** button to execute the example.

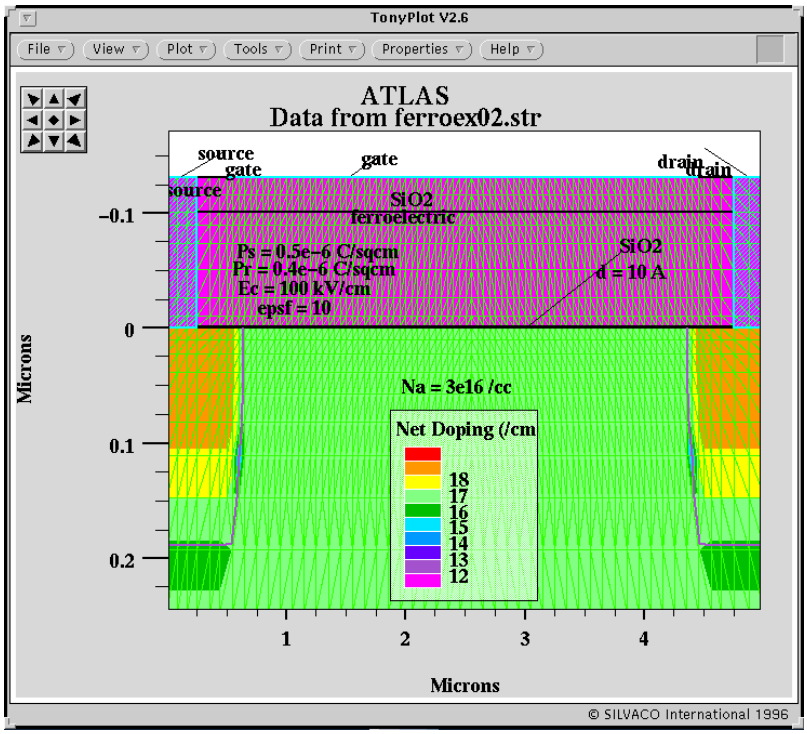


Figure 22.7: 2D MOS Device with Oxide/Ferroelectric/Oxide Gate insulator

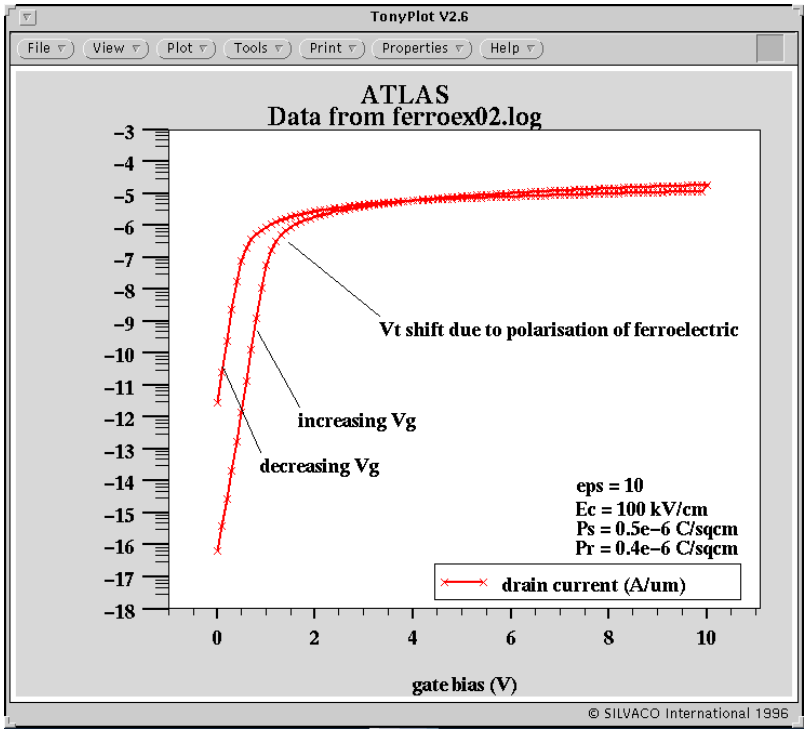


Figure 22.8: Hysteresis effect in the MOS ID/VG curve due to ferroelectric polarization

Input File ferro/ferroex02.in:

```
1    go atlas
```

```
# ferroelectric fet example
# demonstrates Vt shift due to ferroelectric polarization
#
# Ref: Miller, S.L. and McWhorter, P.J., "Physics of the Ferroelectric
# Non-Volitile Memory  Field Effect Transistor", J. Appl. Phys. 72(12)
# 15 Dec. 1992, pp. 5999-6010.
#
# mesh and structure definition
#
mesh space.mult=1.0 outf=ferroex02.str
x.m l=0    s=0.1
x.m l=0.25 s=0.1
x.m l=0.5  s=0.05
x.m l=4.5  s=0.05
x.m l=4.75 s=0.1
x.m l=5.0  s=0.1

y.m l=-0.131 s=0.015
y.m l=-0.101 s=0.015
y.m l=-0.051 s=0.025
y.m l=-0.001 s=0.01
y.m l=0.0  s=0.01
y.m l=1.0  s=0.2

eliminate y.dir y.min=0.2 x.min=0.5 x.max=4.5
eliminate y.dir y.min=0.4

region num=1 silicon y.min=0.0
region num=2 oxide y.max=0.0 y.min=-0.001
region num=3 oxide y.max=-0.001 y.min=-0.101
region num=4 oxide y.max=-0.101 y.min=-0.131

electrode num=1 name=source y.max=0.0 x.max=0.25
electrode num=2 name=drain y.max=0.0 x.min=4.75
electrode num=3 name=gate top x.min=0.5 x.max=4.5
electrode num=4 name=substrate bottom

doping uniform conc=3e16 p.type
doping gaus conc=1e19 n.type x.right=0.5 char=0.075
doping gaus conc=1e19 n.type x.left=4.5 char=0.075

#
# define ferroelectric material
```

```
45 #
46 model region=3 ferro
47 material region=3 ferro.ps=0.5e-6 ferro.pr=0.4e-6 ferro.ec=100000.0 fer-
     ro.eps=10.0
48
49 #
50 # gate workfunction and channel mobility
51 #
52 contact num=3 n.poly
53 material mun=800.0
54
55 #
56 # obtain Id - Vg characteristic
57 #
58 method newton autonr
59 solve
60 solve vd=0.1
61 #
62 # forward sweep
63 #
64 log outf=ferroex02.log
65 solve vgate=0.0 vstep=0.1 vfinal=10.0 name=gate
66 #
67 # reverse sweep
68 #
69 material ferro.ec=-100000.0
70 solve vstep=-0.1 vfinal=0.0 name=gate
71
72 tonyplot -st ferroex02.str -set ferroex02_0.set
73 tonyplot -st ferroex02.log -set ferroex02_1.set
74 quit
75
76
```