

# **SIMDET2016**

## **Introduction to the SILVACO TCAD**

de lundi 5 septembre 2016 à 13:00 à mercredi 7 septembre  
2016 à 13:00 (Europe/Paris)  
à LPNHE Paris ( 1222-RC-08 )  
4 place Jussieu 75005 Paris

## Hands-on session: SILVACO

### Contents Overview

1. Silvaco TCAD input deck layout
2. Optoex01 – Introduction to basic syntax and play with doping & width.
3. Optoex02 – Experiment with DC and AC bias conditions.
4. Optoex03 – Experiment with beam settings, beam power, DC, ramp up and ramp down.
5. Optoex05 - Modify spectral response range.
6. Optoex07 – Experiment with c-function for photogeneration.
7. Mos1ex01 - Introduce 2D structures. Process and device simulation. Substitute variables. Use DevEdit3d to transform 2D  $\Leftrightarrow$  3D.
8. .mesfetex04 : Deep Level Bulk Traps – DC/AC/Transient Analysis.
9. Radex03 - SEU in 3D - Experiment with location and angle of strike.
10. Radex05 - Experiment with c-function for charge generation.
11. Radex07 - Total dose effect, radiation statement. Features in Victory device.

## 1. Silvaco TCAD input deck layout

### Creating a simulation deck in Atlas – Statement Order

<b>Group</b>	<b>Statements</b>
Structure Specification	MESH REGION ELECTRODE DOPING
Material & Model Specification	MATERIAL MODELS CONTACT INTERFACE
Numerical Methods	METHOD
Electro-Optical Solution	LOG SOLVE LOAD SAVE
Results Analysis	EXTRACT TONYPLOT

N.B. This list is not exhaustive. Also, not every statement outlined above needs to be used.

## 2. Optoex01 – Introduction to basic syntax

This is an example of a PIN photodetector. The extraction techniques used on the PIN device will equally apply to other photodetector devices.

The first section of the input file specifies the mesh. In this case, the mesh is 10 microns by 10 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  $1.0 \times 10^{14}$ , 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.

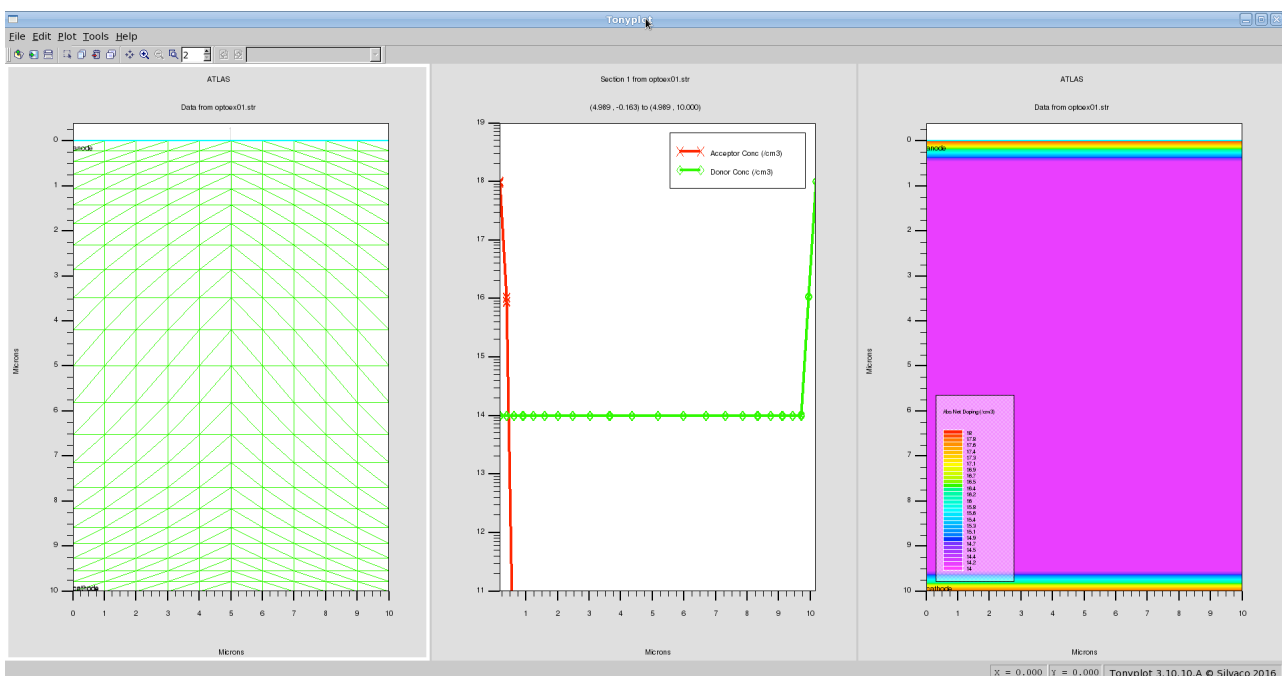


Figure 1: Device mesh, doping profile in 1D & 2D.

## 3. Optoex02 – Experiment with DC and AC bias conditions.

Deck demonstrates the characterization of steady state bias solution of the device, and demonstrates extraction of DC currents as a function of optical source power. It also demonstrates the use of multi-spectral sources. The source spectrum is described in the file "**optoex02.spec**".

The first and second sections are similar to the previous example.

The third section of the input file specifies the optic source. In this example the source originates 1 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 5 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<sup>2</sup>. 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, source photo current, is the equivalent current what 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.

We can experiment with small signal AC analysis on the solve statement

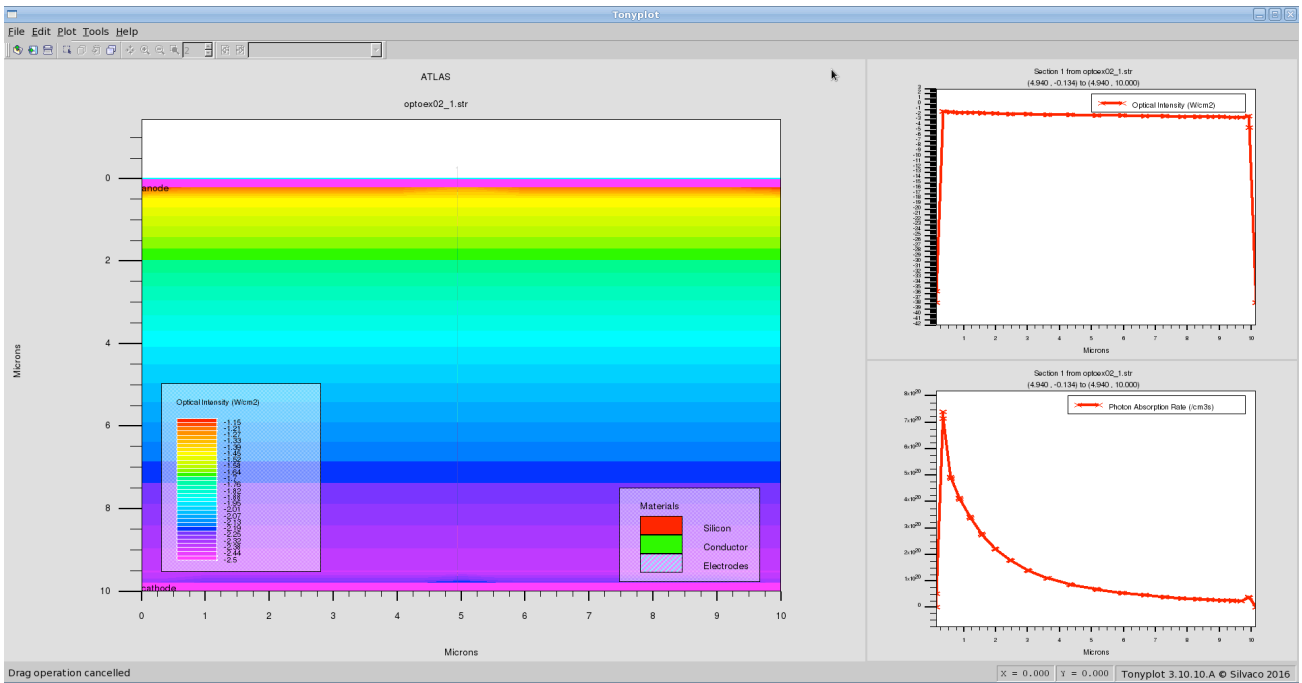


Figure 2 showing the optical intensity & absorption profile.

## **4. Optoex03 – Experiment with beam settings, beam power, DC, ramp up and ramp down.**

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. The first section specifies that the device structure is read in from an external file. The second section specifies the material models used. 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  $5 \text{ W/cm}^2$  is obtained. Then, the source intensity is ramped linearly to 0 over a time of 1 ns. Terminal data is then collected over the next 9 ns. The terminal data is saved to the log file.

The results 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 a nanosecond.

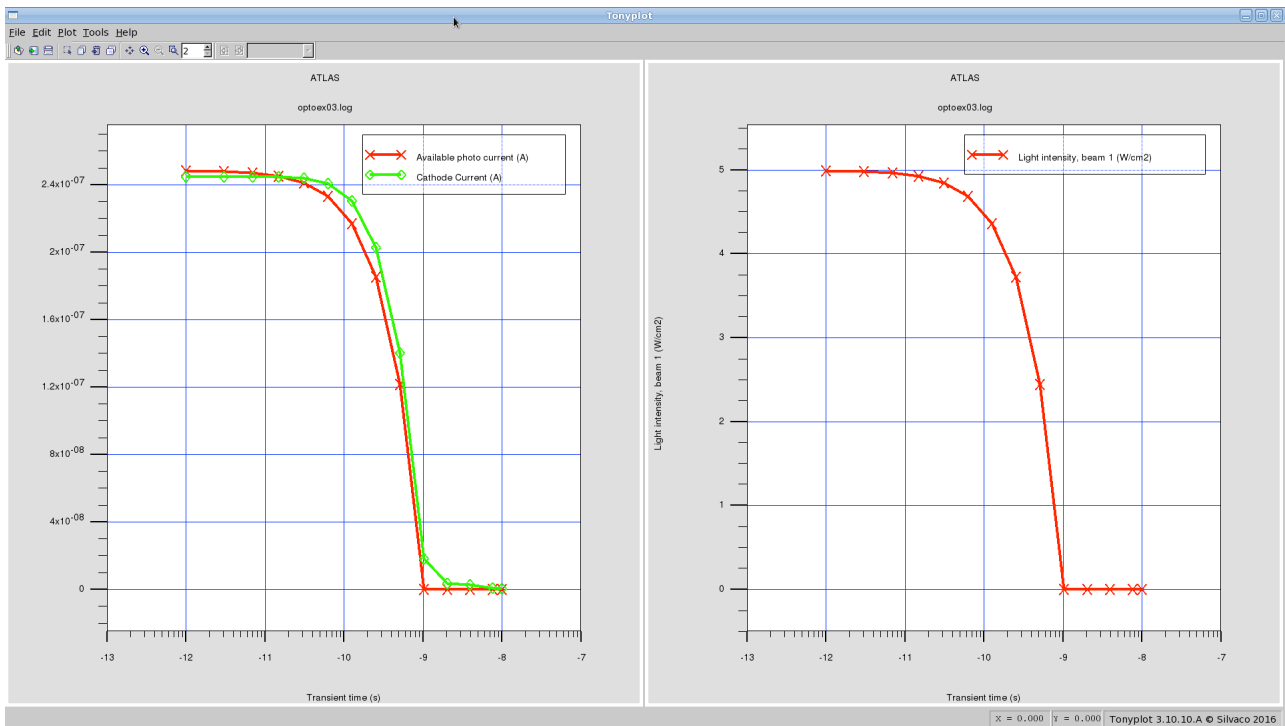


Figure 3 shows photocurrent and light intensity behaviour with time.



## 5. Optoex05 - Modify spectral response range.

This example demonstrates the extraction of spectral response data.

In this deck a monochromatic wavelength of 100nm is specified. In the last section of the file, solutions are obtained for a 1 W/cm<sup>2</sup> 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.

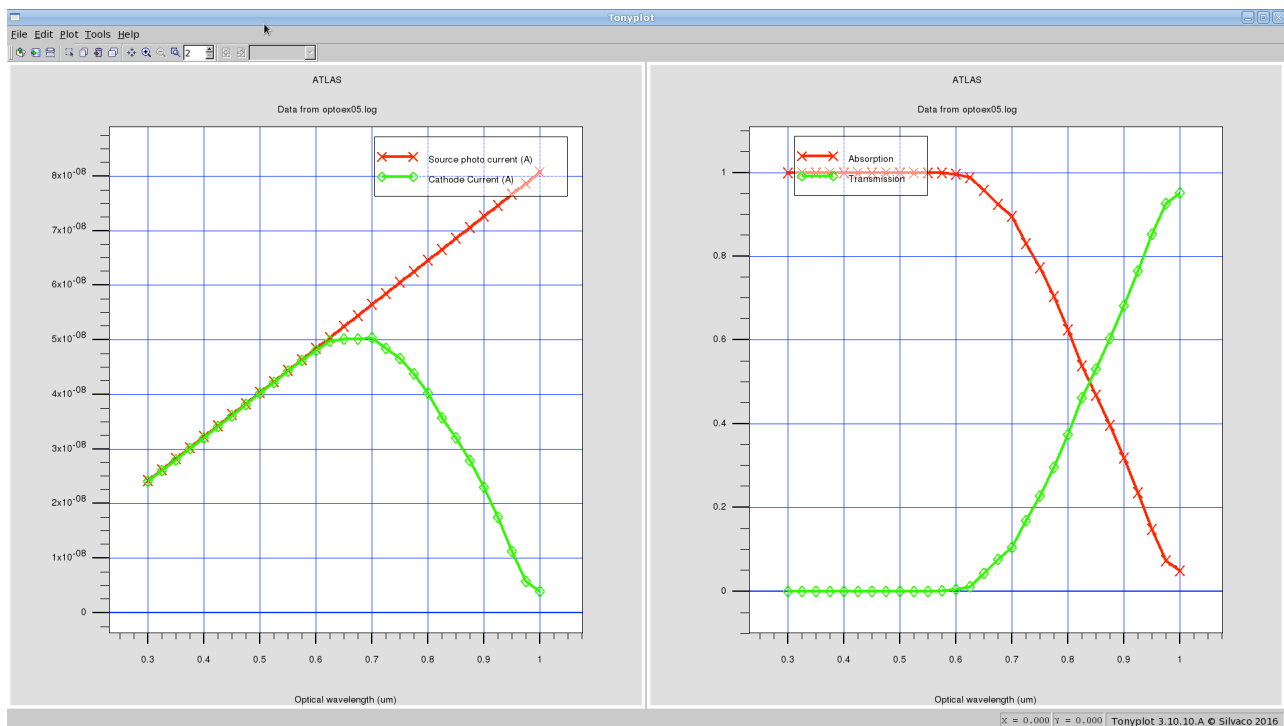


Figure 4 shows the spectral response of the diode. Left hand side pane shows the quantum efficiency with wavelength.

## 6. Optoex07 – Experiment with c-function for photogeneration.

This example demonstrates how an C 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 function is 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** and 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 is being used in the first case.

The user-defined function is in the text file **optoex07.lib** . 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 of the **solve** statement.

The C-functions looks like:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <ctype.h>
#include <malloc.h>
#include <string.h>
#include <template.h>

/*
 * -----
 * ATLAS Parser Function Template
 * ATLAS Version 5.2.0.R
 * c 1993 - 2000 SILVACO International.
 * All rights reserved.
 * -----
 */

/*
 * Generation rate as a function of position
 * Statement: BEAM
 * Parameter: F.RADIATE
 * Arguments:
 * x          location x (microns)
 * y          location y (microns)
 * t          time (seconds )
 * *rat      generation rate per cc per sec.
```

```
*/  
int radiate(double x,double y,double z,double t,double *rat)  
{  
    double alpha, dum;  
    alpha = 5219.459468;  
    dum = 0.6019e-6/(3.0e8*6.626e-34)*alpha*exp(0.0-alpha*y*1e-4);  
    *rat = dum;  
    return(0);  
}
```

## 7. Mos1ex01 - Introduce 2D structures. Process and device simulation, substitute variables and use DevEdit3d to transform 2D to 3D.

Basic MOS Athena to Atlas interface example simulating an  $I_d/V_{gs}$  curve and extracting threshold voltage and other SPICE parameters. This example demonstrates:

- Process simulation of a MOS transistor in Athena
- Process parameter extraction (eg. oxide thicknesses)
- Autointerface between Athena and Atlas
- Simple  $I_d/V_{gs}$  curve generation with  $V_{ds}=0.1V$
- Parameter extraction for  $V_t$ , linear gain (beta) and mobility rolloff (theta)

The process simulation follows a standard LDD MOS process. The process steps are simplified and default models are used to give a fast runtime. The polysilicon gate is formed by a simple geometrical etch. Before this point the simulation is essentially one dimensional and hence is run in Athena's 1D mode. After the poly etch, the structure converts to 2D.

The grid used in this example is defined quite tightly. However the statement **init ... spac.mult=3** relaxes the mesh in X and Y directions by a factor of three. A more typical mesh for MOS simulation can be obtained by setting **spac.mult=1** .

Using DeckBuild's auto-interface, the process simulation structure will be passed into Atlas automatically. This auto-interface therefore allows global optimization from process simulation to device simulation to SPICE model parameter extraction.

The **extract** statement at the end of the file is used to calculate the oxide thickness at that point. The value returned here may be used as an optimization target for calibration. This value will be appended to a file in the current working directory called **results.final**. Significant use of **extract** statements is strongly recommended, especially at points where in-line Fab measurements are taken.

Electrodes are defined at the end of the process simulation. Metal is deposited and patterned. Then **electrode** statements are used to define the metal regions plus the polysilicon as electrodes for use in Atlas.

In Atlas the first task is to define the models and material parameters for the simulation. The **contact** statement is used to define the workfunction of the gate electrodes, while the **interface** statement defines the fixed charge at the silicon/oxide interface. For simple MOS simulation the parameters CVT and SRH define the recommended models. **CVT** sets a general purpose mobility model including concentration, temperature, parallel field and transverse field dependence.

The statement **solve init** is used to solve the thermal equilibrium case. After this the voltages can be ramped. It is recommended to use small steps at first when ramping voltages. Once two non-zero biases have been obtained, the program uses a projection as initial guesses to further bias points. The projection method allows larger voltage steps to be taken. The syntax **method trap** enables Atlas to cut user-defined voltage steps in half if convergence is not obtained. This is a highly recommended option and is turned on by default.

The unique feature of this example is the IV data simulated and the extraction syntax used. The model, interface and contact statements in Atlas are also as in the previous example.

The sequence of **solve** statements is set to ramp the gate bias with the drain voltage at 0.1V. Solutions are obtained at 0.25V intervals up to 3.0V. All terminal characteristics are saved to the file mos1ex02\_1.log as specified in the **log** statement.

The **extract** statements at the end of the file are used to measure the threshold voltage and other SPICE parameters. The results from the extract statements are printed in the run-time output, saved to a file called results.final and optionally used in the optimizer or VWF automation tools. The syntax used in these statements is freely composed of operators such as maximum value (max) and simulation results such as drain current (i."drain"). The **name** parameter specifies only a user-defined label. The routines are not hard-coded to these names. Thus the first extraction statement reads: extract the value called nvt found by taking the x intercept of the maximum slope to the curve of drain voltage vs. drain current and subtracting half the drain voltage. This is just one possible definition of threshold voltage. Current search methods are also possible and are described under the DIBL example later in this section. The VWF interactive tools manual has details of the extract syntax.

The second extract statement measures the gain (or Beta). This is defined as the value of the steepest slope to the  $I_d/V_{gs}$  curve divided by the drain voltage. The final extraction is for the SPICE level 3 mobility roll-off parameter (or Theta). This syntax shows the use of

the syntax: `$$nvt` and `$$nbeta` . This tells DeckBuild to substitute the previously extracted values of threshold and beta into this places in the equation.

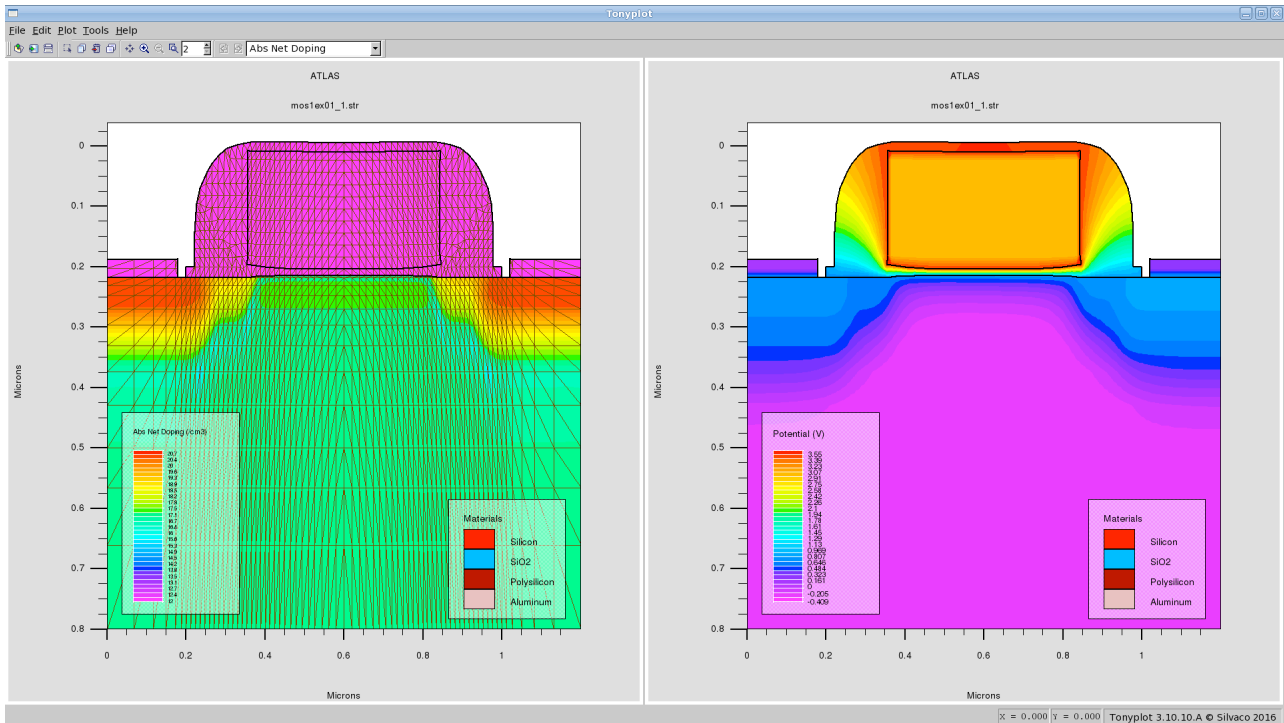


Figure 5 shows the structure mesh, doping and potential contours.

## 8. mesfetex04 : Deep Level Bulk Traps – DC/AC/Transient Analysis.

This examples demonstrate the simulation of trap states. The example consists of several sections:

- Formation of a MESFET structure and doping using ATHENA
- DC simulation of  $I_d/V_{gs}$  characteristics including traps
- Transient simulation of gate turn off including traps
- Frequency domain AC simulation including traps
- S-parameter extraction including traps

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  $0.1\mu m$  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 performed. 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 that a shift to a more positive threshold voltage is caused by the presence of the donor traps.

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, whilst 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 program. A `dt` value of `ramptime/100` is a 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`.

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

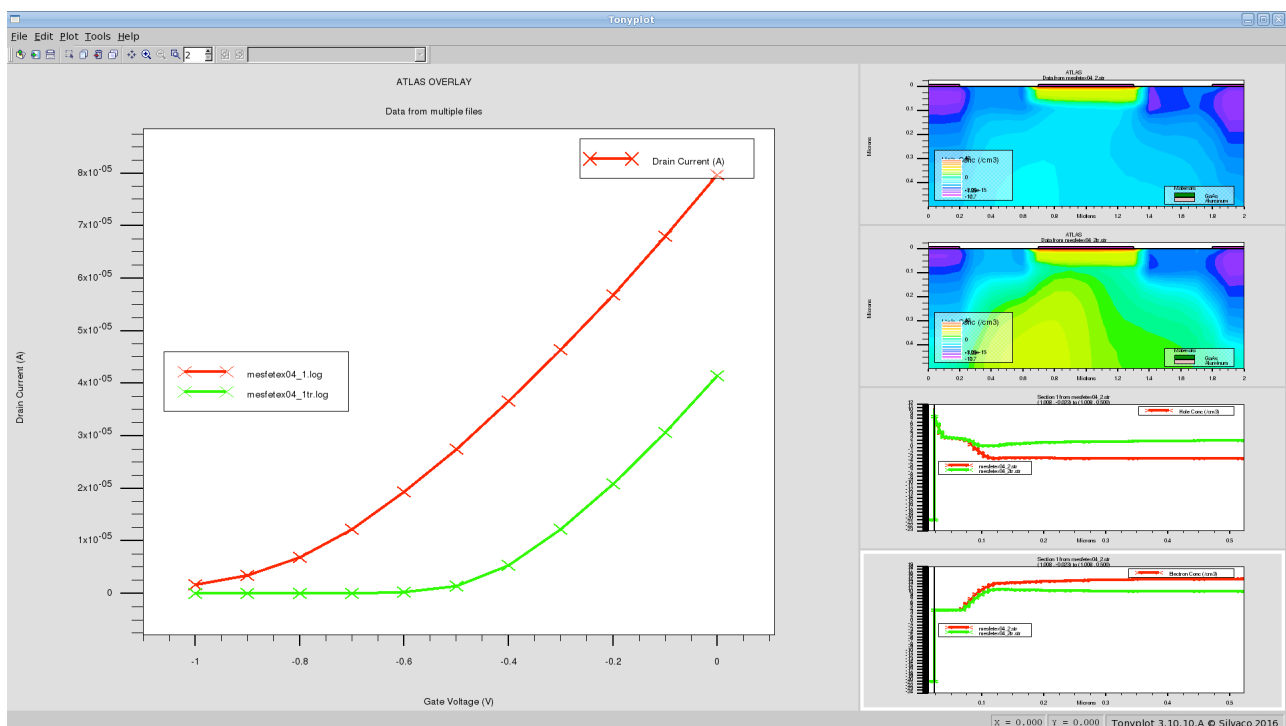


Figure 6 shows the effect of traps on the hole and electron concentration profile.



## 9. Radex03 - SEU in 3D - Experiment with location and angle of strike.

This example demonstrates simulation of the effects of different angles of incidence of an SEU in a MOSFET structure. The MOSFET structure is constructed using DevEdit 3D. The structure is then passed to Atlas for electrical testing. The input file consists of the following parts:

- Construction of the device in DevEdit 3D
- 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 the users to add their own position and time dependent charge generation using the C-Interpreter and incorporate these into Atlas simulations.

DevEdit 3D generates 2 types of files: DevEdit 3D input file and the structure file. The first can be run in the 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 DevEdit 3D 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 DevEdit 3D input files can also be read directly into the graphical user interface of DevEdit 3D to provide all the menu options used to construct the structure.

The Atlas simulation begins by reading the structure from DevEdit 3D. DeckBuild provides an autointerface between DevEdit 3D and Atlas so that the structure produced by DevEdit 3D is transferred to Atlas without having to indicate the **mesh** statement (commented out in this example). Without the automatic DevEdit 3D/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 model 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 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, and 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 **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 this example's 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  $1.e18 \text{ cm}^{-3}$ . 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 ( $t=4 \text{ ps}$ ).

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  $V_{ds}=5 \text{ V}$  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.

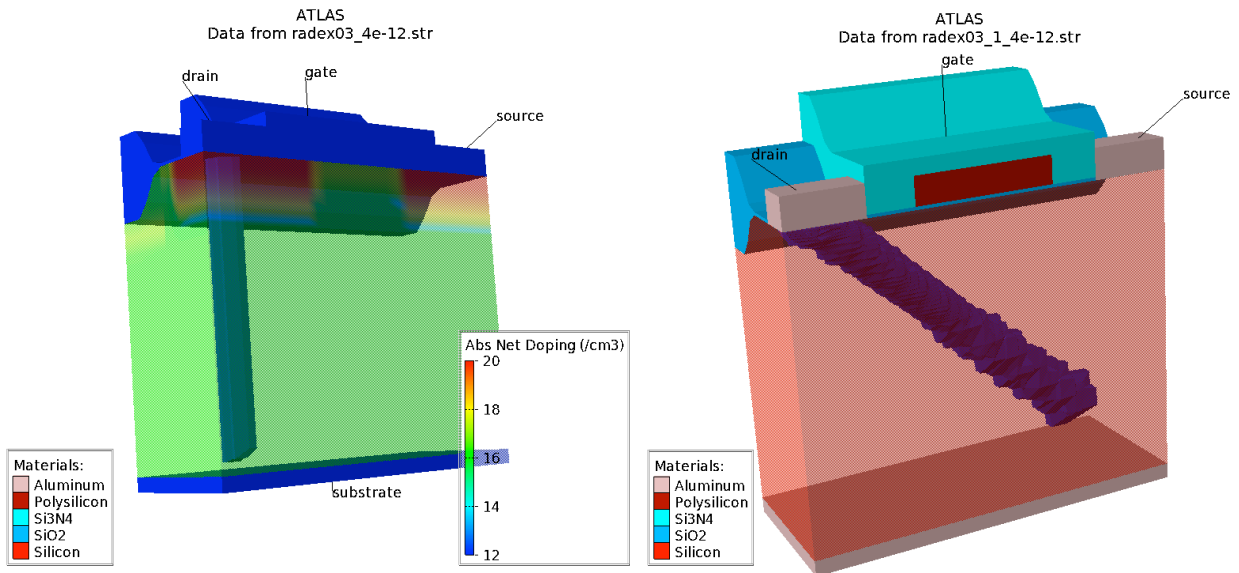


Figure 7 shows the SEU generation rate isosurface along the direction of the ionizing particle.

## 10. Radex05 - Experiment with c-function for charge generation.

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.

The c-function for the generation rate as a function of locations is shown here:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <ctype.h>
#include <malloc.h>
#include <string.h>
#include <template.h>

/*
 * -----
 --
 * ATLAS Parser Function Template
 * ATLAS Version 5.2.0.R
 * c 1993 - 2000 SILVACO International.
 * All rights reserved.
 * -----
 --
 */

/*
 * Generation rate as a function of position (3D)
 * Statement: BEAM
 * Parameter: F3.RADIATE
 * Arguments:
 * x          location x (microns)
 * y          location y (microns)
 * z          location y (microns)
 * t          time (seconds )
 * *rat       generation rate per cc per sec.
 */
int radiate3(double x, double y, double z, double t, double *rat)
{
    double a, c;
    a = pow((x-5.0)*(x-5.0)+(y-5.0)*(y-5.0),0.5);
    if(a < 2.5)
        c=1.0e22;
    else
```

```

    c = 0.0;
    *rat = c;
    return(0);
}

```

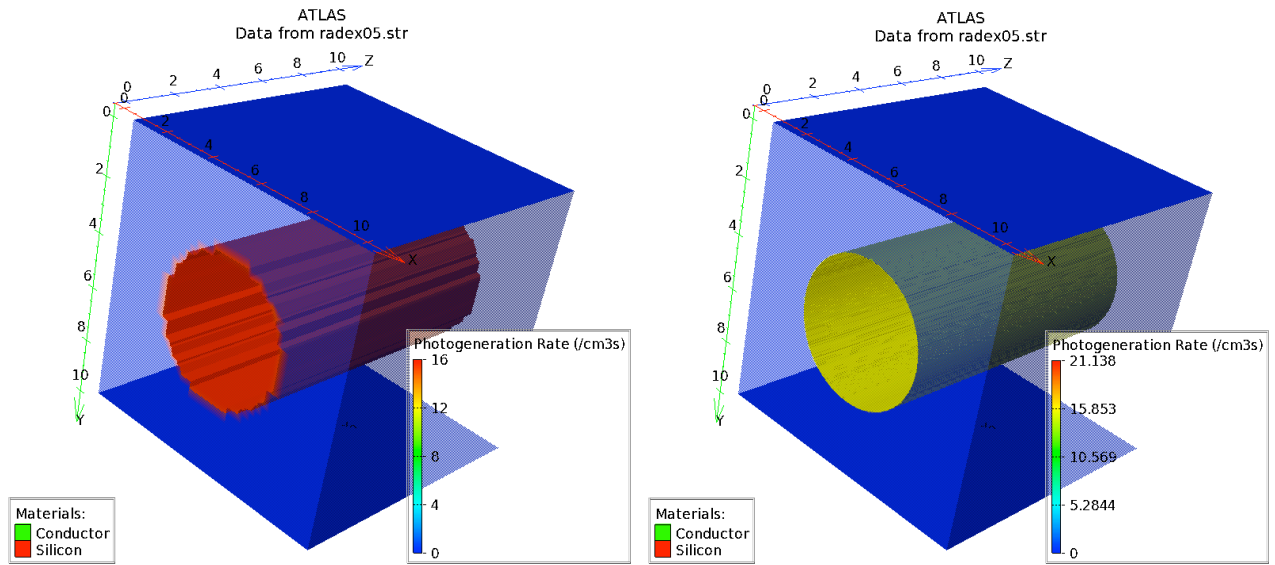


Figure 8 shows the charge generation using simple and customised meshing.

## 11. Radex07 - Total dose effect, radiation statement.

This examples demonstrates positive charge build up and **radiation induced charge de-trapping** in an n-MOS gate oxide as a result of **10keV X-Ray irradiation**.

A physical effect missing in previous total dose models is radiation induced de-trapping of previously trapped oxide charge, as a result of the **trapped charge being released after absorbing energy quanta from the geminate recombination process**. Geminate recombination occurs when an electron-hole pair created from absorption of radiation, recombine again before an electric field can accelerate the pair in opposite directions. Clearly the probability of geminate recombination increases with reducing electric field. Thus **the rate of hole de-trapping in the oxide also increases with a reduction in electric field**, as discussed in the reference for this model from Kimpton et al., "**A Simple Trap-Detrap Model for Accurate Prediction of Radiation Induced Threshold Voltage Shifts in Radiation Tolerant Oxides, for All Static or Time Variant Oxide Fields**" Solid-State Electronics, Vol. 37, No. 1, pp.153-158, 1994

To demonstrate both the radiation induced hole trapping and de-trapping phenomenon, a positive bias is applied to the gate of an n-MOSFET and is irradiated with 10keV X-Rays. Positive oxide charge builds up during irradiation as is monitored using the **probe material=oxide p.conc** statement and by comparing the **pre and post irradiation threshold voltage**. After irradiation to 1MRad, the gate bias is changed to -0.9 volts to minimise the gate oxide field. The device is then re-irradiated with another 1MRad dose and the trapped hole concentration in the gate oxide is monitored.

The greatly reduced oxide field resulting from the change in gate bias increases the geminate recombination process which also greatly increases the rate of hole de-trapping in the oxide, such that it now becomes even greater than the hole trapping rate. It can be seen from this example that rather than increasing the trapped hole concentration with radiation dose, now all the additional energy released from the increased rate of gemnate recombination, now decreases the trapped hole concentration with radiation dose. It should be emphasized here that the **hole de-trapping is NOT a result of hole recombination with electrons, but is a direct result of trapped holes absorbing energy quanta from the geminate recombination process**.

Finally, after the second irradiation with low electric field on the gate, the threshold voltage is simulated again, and shows a recovery from the shift observed after the first irradiation with a high gate oxide field.

To simulate total dose effects in insulators, the oxides are converted to large bandgap semiconductors with the **material material=oxide semiconductor** statement. Then the concentrations and characteristics of the bulk oxide and interface traps are defined in accordance with the referenced paper above, using the **oxidecharging** and **intoxidecharging** statements respectively. Finally the characteristics of the radiation are defined using the **radiation doserate=1 XRay** statement. Then on any transient solve statement the parameter "radiation" is specified to invoke the time dependent irradiation process.

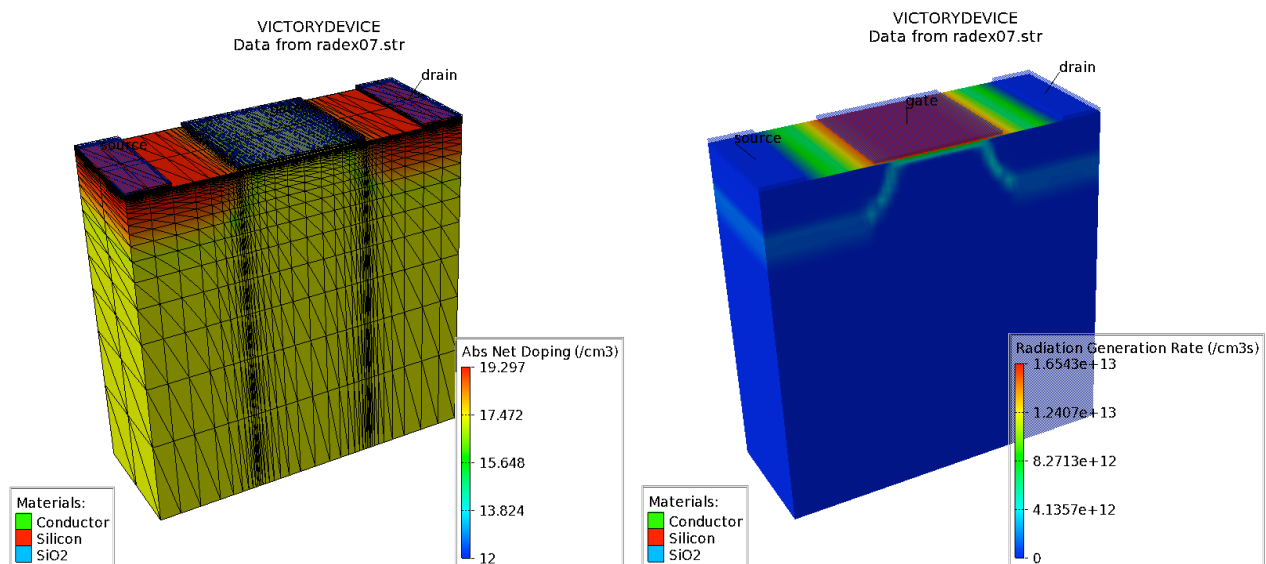


Figure 9 showing n-MOS doping & mesh as well as the radiation generation rate following 1MRad 10KeV X-Ray.

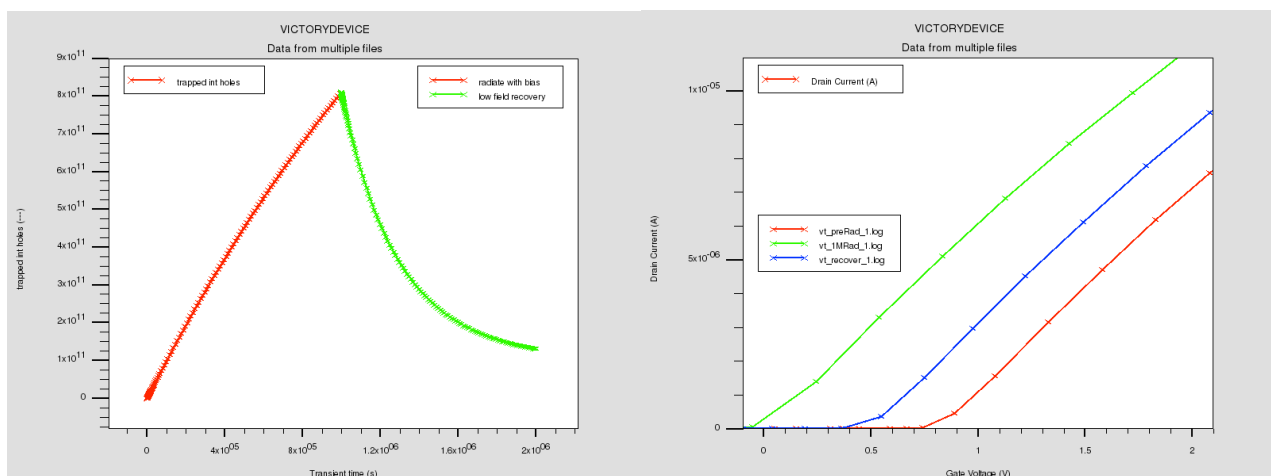


Figure 10 showing the trapping and detrapping of holes as well as the shift and recovery of the threshold voltage following a further radiation at low gate voltage.



## Appendix – Example input decks

### Optoex01.in

```
# (c) Silvaco Inc., 2015
go atlas
Title PIN photodiode simulation example
#
# PIN device description and initial solution
#
# SECTION 1: Mesh Specification
#
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
#
# SECTION 4: Initial Solution
#

solve      init outf=optoex01.str master

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



# Optoex02.in

```
# (c) Silvaco Inc., 2015
go atlas
Title PIN diode DC solutions
#
# PIN DC solution

# PIN device description and initial solution
#
# SECTION 1: Mesh Specification
#
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
#
# 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
power.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
```

## Optoex03.in

```
# (c) Silvaco Inc., 2015
go atlas
Title Light Turn-Off Transient for PN diode
#
# Time Domain Photogeneration Simulation
#
# SECTION 1: Mesh Specification
#
mesh space.mult=1.0
#
x.mesh loc=0.0 spacing=2.5
x.mesh loc=10.0 spacing=2.5
#
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=optoex03_0.str master

tonyplot  optoex03_0.str -set optoex03_0.set

#
# 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=1e-12

tonyplot  optoex03.log -set optoex03_1.set

quit
```

## optoex05.in

```
# (c) Silvaco Inc., 2015
go atlas
Title PIN photodiode spectral response
#
# PIN device spectral response.
#

# SECTION 1: Mesh Specification
#
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
#
# SECTION 4: Initial Solution
#

solve      init outf=optoex05.str master

tonyplot  optoex05.str -set optoex05_0.set

#
# SECTION 5: Optical source specification
```

```
#
beam      num=1 x.origin=5.0 y.origin=-1.0 angle=90.0 wavelength=.1
#

method newton trap
solve     init
solve     vcathode=0.2
#
# SECTION 5: Spectral Response
#
log outf=optoex05.log master
solve prev b1=1 lambda=0.1
solve prev b1=1 lambda=0.2
solve prev b1=1 lambda=0.3
solve prev b1=1 lambda=0.4
solve prev b1=1 lambda=0.5
solve prev b1=1 lambda=0.6
solve prev b1=1 lambda=0.7
solve prev b1=1 lambda=0.8

tonyplot  optoex05.log -set optoex05_1.set
quit
```

## optoex07.in

```
# (c) Silvaco Inc., 2015
go atlas
Title Radiation interpreter function example
#
# Radiation interpreter function example
#
# SECTION 1: Mesh Specification
#
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
```



## mos1ex01.in

```
# (c) Silvaco Inc., 2015
go athena

#
line x loc=0.0 spac=0.1
line x loc=0.2 spac=0.006
line x loc=0.4 spac=0.006
line x loc=0.6 spac=0.01
#
line y loc=0.0 spac=0.002
line y loc=0.2 spac=0.005
line y loc=0.5 spac=0.05
line y loc=0.8 spac=0.15
#
init orientation=100 c.phos=1e14 space.mul=2

#pwell formation including masking off of the nwell
#
diffus time=30 temp=1000 dryo2 press=1.00 hcl=3
#
etch oxide thick=0.02
#
#P-well Implant
#
implant boron dose=8e12 energy=100 pears

#
diffus temp=950 time=100 weto2 hcl=3
#
#N-well implant not shown -
#
# welldrive starts here
diffus time=50 temp=1000 t.rate=4.000 dryo2 press=0.10 hcl=3
#
diffus time=220 temp=1200 nitro press=1
#
diffus time=90 temp=1200 t.rate=-4.444 nitro press=1
#
etch oxide all
#
#sacrificial "cleaning" oxide
diffus time=20 temp=1000 dryo2 press=1 hcl=3
#
etch oxide all
#
#gate oxide grown here:-
```

```
diffus time=11 temp=925 dryo2 press=1.00 hcl=3
#
# Extract a design parameter
extract name="gateox" thickness oxide mat.occno=1 x.val=0.05

#
#vt adjust implant
implant boron dose=9.5e11 energy=10 pearson

#
depo poly thick=0.2 divi=10
#
#from now on the situation is 2-D
#
etch poly left p1.x=0.35
#
method fermi compress
diffuse time=3 temp=900 weto2 press=1.0
#
implant phosphor dose=3.0e13 energy=20 pearson
#
depo oxide thick=0.120 divisions=8
#
etch oxide dry thick=0.120
#
implant arsenic dose=5.0e15 energy=50 pearson
#
method fermi compress
diffuse time=1 temp=900 nitro press=1.0
#

# pattern s/d contact metal
etch oxide left p1.x=0.2
deposit alumin thick=0.03 divi=2
etch alumin right p1.x=0.18

# Extract design parameters

# extract final S/D Xj
extract name="nxj" xj silicon mat.occno=1 x.val=0.1 junc.occno=1

# extract the N++ regions sheet resistance
extract name="n++ sheet rho" sheet.res material="Silicon"
mat.occno=1 x.val=0.05 region.occno=1

# extract the sheet rho under the spacer, of the LDD region
extract name="ldd sheet rho" sheet.res material="Silicon" \
    mat.occno=1 x.val=0.3 region.occno=1

# extract the surface conc under the channel.
extract name="chan surf conc" surf.conc impurity="Net Doping" \
```

```
material="Silicon" mat.occno=1 x.val=0.45

# extract a curve of conductance versus bias.
extract start material="Polysilicon" mat.occno=1 \
    bias=0.0 bias.step=0.2 bias.stop=2 x.val=0.45
extract done name="sheet cond v bias" \
    curve(bias,ldn.conduct material="Silicon" mat.occno=1
region.occno=1)\
    outfile="extract.dat"

# extract the long chan Vt
extract name="nldvt" ldvt ntype vb=0.0 qss=1e10 x.val=0.49

structure mirror right

electrode name=gate x=0.5 y=0.1
electrode name=source x=0.1
electrode name=drain x=1.1
electrode name=substrate backside

structure outfile=moslex01_0.str

# plot the structure
tonyplot moslex01_0.str -set moslex01_0.set

##### Vt Test : Returns Vt, Beta and Theta
#####
go atlas

# set material models
models cvt srh print

contact name=gate n.poly
interface qf=3e10

method newton
solve init

# Bias the drain
solve vdrain=0.1

# Ramp the gate
log outf=moslex01_1.log master
solve vgate=0 vstep=0.25 vfinal=3.0 name=gate
save outf=moslex01_1.str

# plot results
tonyplot moslex01_1.log -set moslex01_1_log.set

# extract device parameters
```

```
extract name="nvt"  
(xintercept(maxslope(curve(abs(v."gate"),abs(i."drain")))) \  
  - abs(ave(v."drain"))/2.0)  
extract name="nbeta"  
slope(maxslope(curve(abs(v."gate"),abs(i."drain")))) \  
  * (1.0/abs(ave(v."drain")))  
extract name="ntheta" ((max(abs(v."drain")) *  
$"nbeta")/max(abs(i."drain"))) \  
  - (1.0 / (max(abs(v."gate")) - ($"nvt")))  
  
quit
```

## mesfetex04.in

```
# (c) Silvaco Inc., 2015
go athena
#
line x loc=0.00 spac=0.10
line x loc=2 spac=0.10
#
line y loc=0.1 spac=0.01
line y loc=0.5 spac=0.1
#
init gaas c.silicon=1.0e11
#
deposit gaas thick=0.10 c.silicon=2.0e17 divisions=15 dy=0.005
ydy=0.00
#
deposit nitride thick=0.5 divisions=4
etch nitride left p1.x=0.2
etch nitride right p1.x=1.8
relax y.min=0.2
#
implant silicon dose=1.0e15 energy=30
etch nitride all
#
deposit alum thickness=0.01 div=1
etch alum start x=0.2 y=-5
etch cont x=0.7 y=-5
etch cont x=0.7 y=5
etch done x=0.2 y=5
etch alum start x=1.3 y=-5
etch cont x=1.8 y=-5
etch cont x=1.8 y=5
etch done x=1.3 y=5
#
electrode name=source x=0.1
electrode name=gate x=1.0
electrode name=drain x=1.9
#
structure outf=mesfetex04_0.str
tonyplot mesfetex04_0.str -set mesfetex04_0.set

go atlas
#
contact name=gate work=4.87
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=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

quit
```

## radex03.in

```
# (c) Silvaco Inc., 2015
go devedit
#
# 3D MOS transistor SEU simulation
#

DevEdit version="2.0" library="1.14"

work.area left=0 top=-0.6 right=1 bottom=2.1

# SILVACO Library V1.14

region reg=1 mat=Silicon color=0xffc000 pattern=0x3 Z1=0 Z2=2.5 \
  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"
#
impurity id=1 region.id=1 imp=Boron color=0x906000 \
  x1=0 x2=0 y1=0 y2=0 \
  peak.value=2e+16 ref.value=0 z1=0 z2=0 comb.func=Multiply \
  rolloff.y=both conc.func.y=Constant \
  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=0xfffff00
pattern=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
pattern=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
pattern=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=0.03 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=0.03 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
#
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=radex03_0.str

go atlas

title SEU of MOSFET: normal incidence

# Load the structure generated in DEVEDIT...
# While running in the deckbuild after DEVEDIT input deck the
structure is
# transfered automatically: no mesh statement is needed.

# Gate contact definition
contact num=1 name=gate workfun=4.1

# Models specification
models srh auger cvt

# Single Event Upset
# Specify the charge track: normal incidence through the drain
```

```
single entry="0.1,0.0,2.25" exit="0.1,2.0,2.25" radius=0.05
density=1.e18 \
    t0=4.e-12 tc=2.e-12

# Specify numerical method
method newton

# Get initial condition of the structure
solve init

# 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=radex03_5.str

# SEU: Calculate transient response
method halfimpl dt.min=1.e-12

log outf=radex03_1.log
solve tfinal=4.e-12 tstep=1.e-12
save outf=radex03_4e-12.str
solve tfinal=1.e-7 tstep=1.e-12

#

go atlas

title SEU of MOSFET: oblique incidence

mesh infile=radex03_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=radex03_5.str

solve vdrain=5.0

# SEU: Calculate transient response

method halfimpl dt.min=1.e-12

log outf=radex03_2.log

solve tfinal=4.e-12 tstep=1.e-12

save outf=radex03_1_4e-12.str

solve tfinal=1.e-7 tstep=1.e-12

# Plot drain current transients for the normal and oblique
particle incidence
tonyplot -overlay radex03_1.log radex03_2.log -set radex03_log.set
tonyplot3d radex03_0.str -set radex03_str_0.set
tonyplot3d radex03_5.str -set radex03_str_1.set
tonyplot3d radex03_4e-12.str -set radex03_str_2.set
tonyplot3d radex03_1_4e-12.str -set radex03_str_3.set
#

quit
```

## radex05.in

```
# (c) Silvaco Inc., 2015
go atlas
Title Radiation interpreter function example
#
# Radiation interpreter function example
#
# SECTION 1: Mesh Specification
#
mesh three.d space.mult=1.0
#
x.mesh loc=0.0 spacing=1.0
x.mesh loc=10.0 spacing=1.0
#
y.mesh loc=0.0 spacing=0.1
y.mesh loc=5.0 spacing=0.2
y.mesh loc=10.0 spacing=0.1
#
z.m l=0 spacing=1
z.m l=5.0 s=1
z.m l=10.0 spacing=1
#
# 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
#
# define a function for the radiation
beam f3.radiate=radex05.lib
#
# SECTION 4: Initial Solution
#

solve      init
#
```

```
# SECTION 5: Current vs. intensity
#
method newton
solve      b1=0.000001 outf=radex05.str master
tonyplot3d radex05.str -set radex05.set

quit
```

## radex07.in

```
go victorydevice simflags="-80"
```

```
#-----
-----
# Structure Definition
#-----
-----
```

```
mesh three.d
```

```
x.mesh location=-2      spacing=0.25
x.mesh location=-1.5    spacing=0.25
x.mesh location=-0.75   spacing=0.02
x.mesh location=0       spacing=0.2
x.mesh location=0.75    spacing=0.02
x.mesh location=1.5     spacing=0.25
x.mesh location=2       spacing=0.25
```

```
y.mesh location=-0.05   spacing=0.02
y.mesh location=-0.013  spacing=0.002
y.mesh location=0       spacing=0.001
y.mesh location=4       spacing=1
```

```
z.mesh location=0       spacing=0.75
z.mesh location=1.5     spacing=0.75
```

```
region number=1 oxide   y.max=0
region number=2 silicon y.min=0
```

```
electrode name=source x.min=-2      x.max=-1.5 y.min=-0.05 y.max=0
electrode name=gate   x.min=-0.75   x.max=0.75 y.min=-0.05 y.max=-
0.013
electrode name=drain  x.min=1.5     x.max=2     y.min=-0.05 y.max=0
electrode name=substrate bottom
```

```
#-----
-----
# Materials and Models
#-----
-----
```

```
# Use thermionic interface model at oxide/silicon interface
interface thermionic
```

```
doping p.type conc=2e17 uniform
```

```
doping n.type conc=2e19 gaussian peak=0 charact=0.3
ratio.lateral=1 \
    x.left=-2 x.right=-1.2
doping n.type conc=2e19 gaussian peak=0 charact=0.3
ratio.lateral=1 \
    x.left=1.2 x.right=2

# Semiconductor properties for oxide
material material=oxide nc300=2e19 nv300=2e19 eg300=9
semiconductor
material material=oxide mun=1 mup=1e-3
material material=oxide m.vthp=1

# Conditions for radiation-induced oxide charging
intoxidecharging r1material=oxide r2material=silicon \
    jmodel.p nt.p=3e12 sigmat.p=1.5e-13 sigman.p=1e-30
sigmaph.p=1.5e-13 \
    jmodel.n nt.n=1e4 sigmat.n=1e-30 sigmap.n=1e-30
sigmaph.n=1e-30 \
    mfp.phonon=0.013

oxidecharging material=oxide \
    jmodel.p nt.p=4e18 sigmat.p=1.5e-13 sigman.p=1e-30
sigmaph.p=1.5e-13 \
    jmodel.n nt.n=1e10 sigmat.n=1e-30 sigmap.n=1e-30
sigmaph.n=1e-30

# Radiation environment
radiation doserate=1 XRay

# Other models and material properties
models material=silicon klasrh klassen fldmob shira bound.trap
print
contact name=gate n.poly

#-----
# Solution
#-----

# Request structure-file output of band parameters and
recombination rates
output band.param con.band val.band charge u.trap \
    e.mobility h.mobility traps traps.ft \
    e.velocity ex.velocity ey.velocity \
    h.velocity hx.velocity hy.velocity

# Probe conditions near the oxide/silicon interface
probe material=oxide int.donor.traps x=0 y=-1e-6 z=0.75
name="trapped int holes"
```



```
probe material=oxide j.hole v.y x=0 y=-0.001 z=0.75
name="hole current"
probe material=oxide vel.hole v.y x=0 y=-0.001 z=0.75
name="hole velocity"
probe material=oxide geminate x=0 y=-0.001 z=0.75 name="hole
yield fraction"
probe material=oxide p.conc x=0 y=-0.001 z=0.75
name="holes"
probe material=oxide field v.y x=0 y=-0.001 z=0.75
name="field"
```

```
# Probe conditions halfway between gate and interface
probe material=oxide donor.traps x=0 y=-0.0065 z=0.75
name="trapped bulk holes"
probe material=oxide p.conc x=0 y=-0.0065 z=0.75
name="midpoint holes"
```

```
# Set Numerics
method pam.gmres climitt=1e-6 maxtraps=10 dt.max=0.0501
```

```
# Initialize the solution
solve init
solve previous
# save outfile=0V.str
```

```
# Put a small bias on the drain
solve vdrain=0.01 previous
solve vdrain=0.05
```

```
# Ramp up the gate bias
log outfile=radex07_0.log
solve radiation dt=0.01 tstop=1 vgate=5 ramptime=1 previous
# save outfile=preRad.str
log off
```

```
# Irradiate up to 1 MRad
method climitt=1e-4 maxtraps=10 tol.time=1e6 dt.max=10000
tstep.incr=1.1
log outfile=radex07_1.log
solve radiation dt=10 tstop=1000000 previous
save outfile=radex07.str
log off
tonyplot3d radex07.str -set radex07_0.set
```

```
# Reverse the gate bias to zero the oxide field
method climitt=1e-4 maxtraps=10 tol.time=1e6 dt.max=0.0501
dt.min=0.00001
log outfile=radex07_2.log
solve radiation dt=0.01 tstop=1000001 vgate=-0.9 ramptime=1
previous
# save outfile=low_field.str
```

```
log off
```

```
# Re-Irradiate at Near Zero Oxide Field (Vg=-0.9V) to Release  
Trapped Oxide Charge
```

```
method climit=1e-4 maxtraps=10 tol.time=1e6 dt.max=10000
```

```
tstep.incr=1.1
```

```
log outfile=radex07_3.log
```

```
solve radiation dt=0.001 tstop=2000000 previous
```

```
# save outfile=recover.str
```

```
log off
```

```
tonyplot radex07_1.log -overlay radex07_3.log -set radex07_1.set
```

```
# Re-check Recovered Vt
```

```
method climit=1e-4 maxtraps=10 tol.time=1e6 dt.max=0.0501
```

```
dt.min=0.00001
```

```
log outfile=radex07_4.log
```

```
solve radiation dt=0.01 tstop=2000001 vgate=5 ramptime=1 previous
```

```
# save outfile=final.str
```

```
tonyplot radex07_0.log -overlay radex07_2.log radex07_4.log -set  
radex07_2.set
```