Introduction

Sentaurus is a very powerful, industry-standard TCAD tool for simulating the fabrication and operation of electronic and photonic devices. Devices like diodes, transistors, MOSFETs, solar cells can be simulated using this software.

The most general approach to simulating a device is to define the device structure, properties of the materials used in the structure, the physics to be considered and finally to choose parameters to be varied so as to understand the behaviour (output parameters) of the device.

Sentaurus uses silicon as the default material. If only silicon is used in the devices to be studied and the default parameter set for silicon is adequate, then the material definition step can be skipped. The following table (Table Intro_01) shows the simulation steps and the corresponding tools and files used for that purpose.

Sentaurus workbench (SWB) hosts these tools including the input and output parameter lists. The values of the input parameters are extracted from the SWB interface while performing an operation with a particular tool.

Table Intro_01: Tools used for Sentaurus

Purpose	Name of the tool	File to edit
Device structure definition	Sentaurus Structure Editor	*sde_dvs.cmd*
Material properties definition		*sdevice.par*
Physics and input definition	Sentaurus Device	*sdevice_des.cmd*
View voltage-current	Inspect	*inspect_ins.cmd*
View 2D/3D distribution	SVisual	**.tdr*

Sentaurus Workbench anchors all these tools, variables and used to run simulations

Logging In and License Activation

For background information on ECE computing services, please see:
http://www.ece.utoronto.ca/lab-support/home-directories/

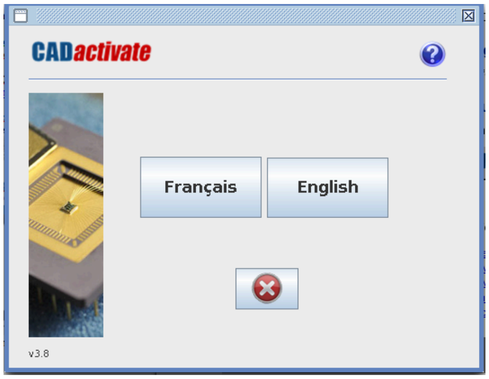
Log into an ug server machine in the lab or remotely.
If you are logging in remotely, type `ssh –X username@ugXXX.eecg.utoronto.ca`
`XXX` = any number in the range of 132-180, or 201-249

The following license instructions are only for late registrants to the course (after August 30, 2017). If you are not a late registrant, your username has already been sent to CMC Microsystems for license use which will allow you to access to the Sentaurus licenses.

1. Open a web browser, such as iceweasel

2. Go to the CMC Microsystems CADactivate page
   http://www.cmc.ca/en/WhatWeOffer/Design/Tools/CADactivate.aspx

   - Click the CADactivate button, and click "Yes" through the warnings about running remote applications. You should then reach this dialog box.

     Choose "English" as your language.
     Agree to account terms.

   - When prompted for the login name and password, please enter the following

     CMC Login name: TRECE335FF2017
     Password: ECE335F!CMC

Steps for the first time …

You have to set the path of Sentaurus in the .cshrc file and copy the sample project. **You only have to do this step once.**

Setting the path

1. Start a Terminal.

   If you are at a workstation in the lab, from the home screen, select Applications > System Tools > MATE Terminal. If you are logging in remotely, you will already have a Terminal screen open.

2. If you don't have a .cshrc file in your home directory (use `ls –a` to see all files), create a ~/.cshrc file in your home directory by following the next two steps. If you already have a .cshrc file, you can edit it in the same way.

3. Type `cd` and press enter to go to your home directory.

4. Type pico .cshrc to open the text editor Pico, and create a file named .cshrc . <u>There is a space between pico and .cshrc</u>

5. Go down of the file until you find a line as

   ```
   # Here's how to add a directory to your shell's path.
   # MH users will want to uncomment this:
   # set path=($path /local/bin/mh)
   ```

6. Add following line in the first blank line after the line mentioned in Step 5

   ```
   source /CMC/scripts/Synopsys.Sentaurus.2013.03
   ```

   <u>There is a space after source and the command is case-sensitive.</u>

7. Enter Ctrl + X to exit. Choose to save the file as .cshrc.

8. Close the terminal and open it again to update the change.

9. If you are logging in remotely, you will need to exit and re-login.

Note: Some students have found that adding the line in Step 6 in the .cshrc file makes them unable to use VNC. If you experience a VNC problem, remove this line from your .cshrc file. Type the line into the prompt after login so you can run Sentaurus TCAD.

Copying Project Files

Before you begin the project, please copy the project files into your directory.

1. Go the directory /cad2/ece335f
    a. At the workstation, choose Applications > System Tools > Caja to go to this directory
    b. Or you can enter `cd /cad2/ece335f` in a Terminal window

2. Right click on the folder `ece335f` and select copy

3. Now go to your home folder again.

4. Now 'Paste' the folder here either by pressing Ctrl + V or right clicking on the empty area of the right panel and choosing 'Paste'.

5. If you are logging in remotely, use the `cp` command to copy the files into your home directory.

Starting Sentaurus

The main module where you control and monitor your simulations is called “Sentaurus Workbench.”

To start Sentaurus workbench, type `swb &` in the terminal window. If you are using remote login and have trouble launching a display, try logging in again or logging into a different workstation.

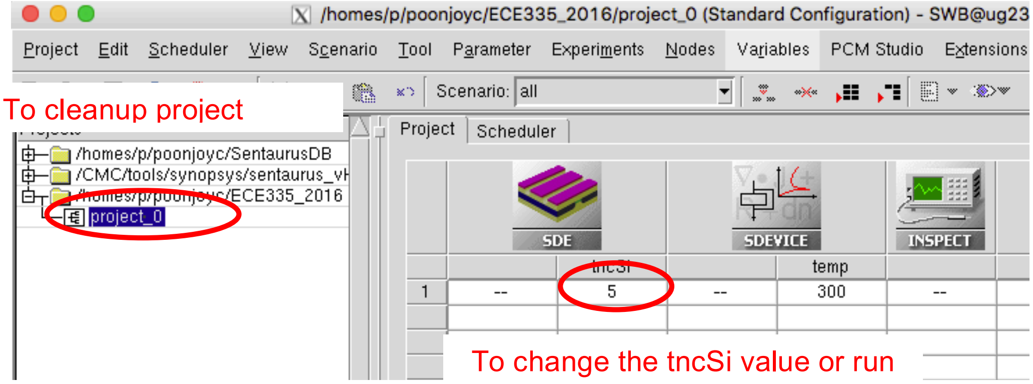
You can open different tools within Sentaurus by going under the “Extensions” in the Sentaurus Workbench and choosing to run the various tools (like Structure Editor, Process, Visualize, etc.). Each tool has its own functionality – e.g., creating a structure to simulate, defining a fabrication process, displaying the simulation results. You will be using these different tools in the project.

A Tour of Sentaurus using Project 0

Once projects are copied in your folder you can explore how to use Sentaurus and the different tools that you can use within it. The following steps are the most common things you would do using Sentaurus. Once familiar with the process, you should be able to proceed on your own.

Clean Up Project Output

1. Right click on the project on the left panel. For now on project_0
2. Select Project > select Clean Up… > Click OK
3. Now all the output of this project will be cleared. Your command files will still be there.


/homes/p/poonjoyc/ECE335_2016/project_0 (Standard Configuration) - SWB@ug23
Project
Edit
Scheduler
View
Scenario
Tool
Parameter
Experiments
Nodes
Variables
PCM Studio
Extensions
Scenario: all
/homes/p/poonjoyc/SentaurusDB
/CMC/tools/synopsys/sentaurus_vH
/homes/p/poonjoyc/ECE335_2016
project_0
Project
Scheduler
SDE
SDEVICE
INSPECT
tncSi
temp
1
--
5
--
300
--

To cleanup project

To change the tncSi value or run

Set a Different Value of tncSi

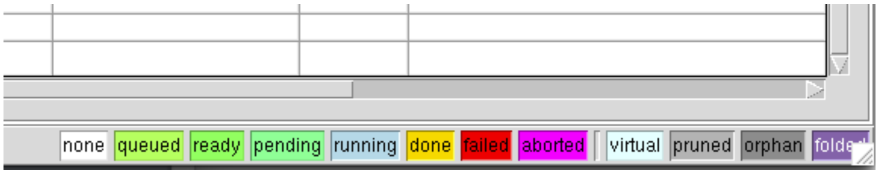
SDE stands for Sentaurus Device Editor, which allows you to vary the parameters (such as the dimensions, doping) of the device.

1. Right click on the value of tncSi that is 5 for now.

2. Select “Edit Value in a Cell” and change the value to 10. So, the thickness of the substrate will be now 10 μm. Or right click and then select “Properties…”, then you can change the value under “Value” column. <u>All units of length for SDE is μm.</u>
3. Click OK.

Run SDE to Create the Structure

1. Right click on value of the last node under SDE that is 10 now. See the above figure.
2. Select Run… Choose Run.
   - Two windows will open. One contains the simulation log, and the second is a window for the Sentaurus Structure Editor, which displays the device.
   - While this node is running, its colour will be light blue in the Sentaurus Workbench. The legend matching the colour and the state of the node is shown in the bottom right corner of the Sentaurus Workbench window.



3. Once the calculation is complete the node will turn yellow indicating its status to be ‘done’.
   - Be patient. Depending on the workstation or server you are using, this step may take a couple of minutes.
4. Close the pop-up window that shows the progress.

Observing the Created Structure

1. Right click on last node of SDE tool again.
2. Select Quick Visualize
3. Wait for Sentaurus Visual to open. When you finish, you may close Sentaurus Visual.
4. On the left side of Sentaurus Visual window, you can change the region and property that you want to view. For example, the default value for scalars is DopingConcentration. You can change it to PhosphorusActiveConcentration. You can also change the Materials, Regions and Lines/KMC.
5. You can do the same when you view the structures from Sentaurus Device.

Observing the Commands for SDE

Once you see the structure you may now be interested to see the command used to create such a structure

1. Select and then right click on the SDE tool icon at the top of the column.

2. Select Edit Input > Commands…
3. This will open sde_dvs.cmd file that is used to define the structure. You can modify the file. If you modify it, to create structure based on your modification, you have to save the file and re-run the SDE tool.
4. To check whether the .cmd file is working properly look for the log file named n8_dvs.cmd.log. 8 is the node number of the last column under sde tool. #f will indicate the failure during execution a given command.

Running Simulation Using SDevice

To simulate the cell you have to run SDevice tool.

1. Right click on the last node under SDevice that is the node that contains the value 300.
2. Select Run and then from the pop-up window select Run again.
3. Wait for simulation to finish and then close the popup window.

Observing the Commands for Sentaurus Device

1. Right click on the tool that is SDevice.
2. Select Edit Input > Commands…

The IV Curve

You have to run Inspect for this purpose.

1. Right click on the Inspect tool icon in the top bar. Select Properties.
2. In the “Tool properties for inspect” window, for “Run as”, select “batch” and then OK.
3. Right click on -- node under Inspect tool.
4. Select Run
5. When finished, right click on the node and choose “Visualize” and then “Inspect (all files)”. Click “yes” to launch Inspect.
6. In the Inspect tool, click “File” and “Load Dataset”. Choose n95_des.plt.
7. Using the left panel you can add more curves
    a. Choose electrode_1 or electrode_2
    b. Choose a parameter, say OuterVoltage
    c. Click on ‘To X-axis’
    d. Choose the electrode and a parameter (e.g., TotalCurrent) for the y-axis and then click on ‘To Y-axis’
8. You can separately open a project output after loading Inspect from Terminal

Observing the Commands for Sentaurus Inspect

1. Right click on the Inspect tool icon
2. Select Edit Input > Commands…

3. Remember the commands for Inspect dictates the figure axis, scale, variables, etc. You need to change the commands if you want to change the figure settings.

Cross-Section Information

1. Right click on the last node of SDevice tool and select Quick Visualize
2. Sentaurus Visual will open and the result of node 95 will be loaded.

   As an alternative to the above steps, you can open Sentaurus Visual by clicking “Extensions” in the top menu of Sentaurus Workbench, and then choosing “Run Sentaurus Visual.” Once in Sentaurus Visual, go to File > Open… and choose n95_des.tdr
3. You will see the list of calculated and defined parameters, such as bandgap, doping concentration, etc. under “Scalars” tab in the second menu on the left.
4. Select “DopingConcentration”
5. To see the cross-section of the “DopingConcenration”, select Tools > Precision Cuts
6. Choose ‘Alignment’ as x.
7. Enter 5 as the first cut point in ‘First Cutline’ field and select ‘Create Cuts’
8. Note that data from the plots in Sentaurus Visual can be exported (Data > Export XY Data…) or saved as a figure (File > Export Plot…)

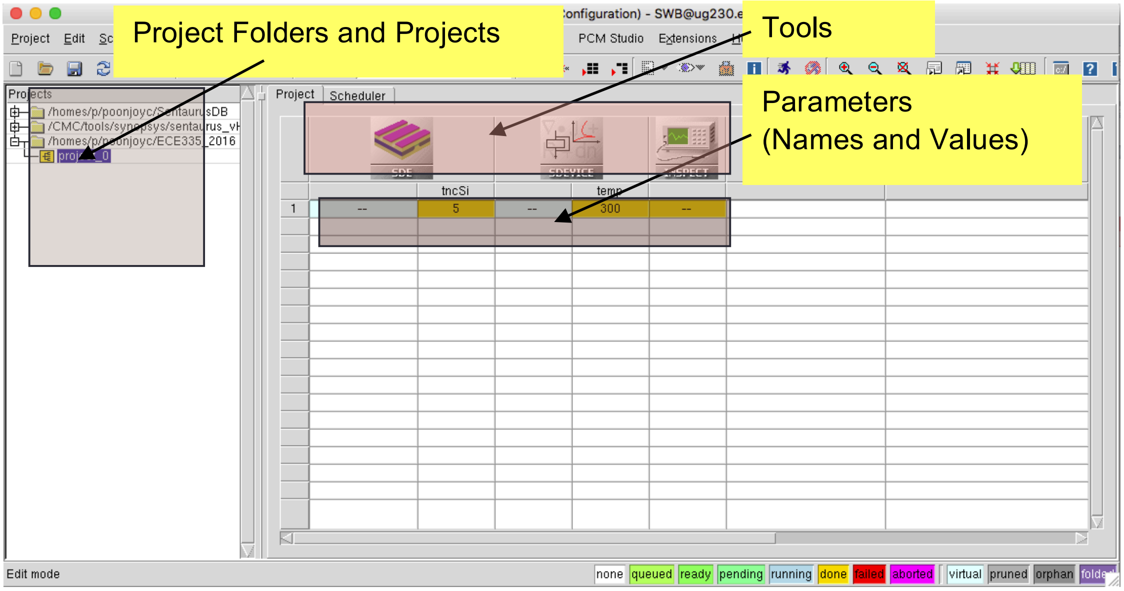
Now you have gone through using Sentaurus starting from creating a structure, simulating it and finally observing the simulated parameters. The next chapters will cover details on the commands used to perform these tasks and the tools.

Sentaurus Work Bench (SWB)

Opening Sentaurus WorkBench

As mentioned before, type swb in the terminal and press enter

The Interface


Project Folders and Projects

Tools

Parameters (Names and Values)

configuration) - SWB@ug230.e
Project Edit Sc
PCM Studio Extensions
Projects
/homes/p/poonjoyc/SentaurusDB
/CMC/tools/synopsys/sentaurus_vH
/homes/p/poonjoyc/ECE335_2016
proj_0
Project Scheduler
SDE
SDEVICE
INSPECT
tncSi
temp
1
--
5
--
300
--
Edit mode
none queued ready pending running done failed aborted virtual pruned orphan folder

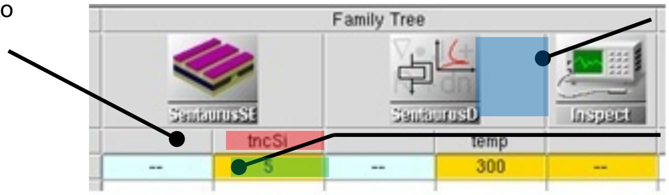
Figure SWB_01: Description of SWB interface

Adding a Parameter

1. Right click the parameter next to which you want add a new parameter
2. Choose' Add'
3. Enter the name of the parameter and its default value
4. Choose whether you want to add the parameter after or before the selected step, and Click OK.

Figure SWB_02 shows where you should click to add a parameter next to tncSi (red region).

Click on the red region to add a parameter after parameter named tncSi or to Add new values of tncSi


Click on the blue region to add a tool, say Inspect

Click on the yellow green zone to change the present value, run SDE

Figure SWB_02: Tools, nodes and Parameters. Figure shows how to change current values and add new one.

Changing Values of a Parameter

1. Double click on the node
2. Change the value and click OK.
3. To change multiple values of a parameter, right click on the parameter name and choose ‘Edit values’.

   Figure SWB_02 shows where you should click to change current value of a parameter (tncSi) (green region).

Adding New Values of a Parameter

1. Right click on the parameter for which you want to add values.
2. Choose ‘Add Values’
3. Choose the value(s) you want to add and click OK

   Figure SWB_02 shows where you should click to add new values of a parameter (tncSi) (red region).

Adding a tool

1. Right click the tool next to which you want add a new parameter
2. Choose’ Add’
3. Click ‘Tools..’ and choose the tool you want to add
4. Choose whether you want to add the tool after or before the selected step and Click OK.

   Figure SWB_02 shows where you should click to add a new tool (blue region).

Running a tool

1. Right click on the last node of a tool
2. Choose ‘Run’

   Figure SWB_02 shows where you should click to run SDE (green region). Similarly you can click on 300 to run SDevice.

Adding a Experiment

1. To set an experiment with all the values you want to set choose ‘Add New Experiment’ from ‘Experiment’ Menu.
2. You can also sort experiment based on parameter values. (Choose ‘Sort Experiments…’ for that)

Sentaurus Structure Editor (SDE)

This is the tool used to define a device structure. Sentaurus primarily simulates two dimensional (2D) structures, but it also allows simulation of 3D structures. Defining an electronic structure can be further reduced to following items

- shapes and dimensions of different regions
- electrode placements
- doping profile or distribution
- grid points or mesh to solve the structure

Grid points are the points for which calculations are performed. Selection of more grid points indicates better representation of a system. But it also means more computational load and as a result it would take long time to simulate the device. Moreover, simulation result does not vary significantly after certain fine grid density. The best strategy to address the grid definition is to put more grid points where it is necessary. More grid points can be placed at an interface between two layers since there is a sudden transition at that region. For a thick material fewer grid points (coarse grid) can be utilized at the middle of the layer.

Structure definition by SDE is explained in the following section for a better understanding of the commands. This example is available as ECE_335_Project_0 and the structure is shown in Figure SDE_01. The font of a text for the following section varies based on purpose of the text. Table SDE_01 gives a description of the fonts used in the following section. All the figures are for explanation purpose only and are not part of the SDE command.

Table SDE_01: Font description of the texts used for the next section

Text Sample	**Font Details**
`sde command`	`Courier New, Font size 9`
Explanation of the command	Arial, Font size 10
`sde command syntax`	*`Courier New, Font size 9, italic`*

Any line starting with a semicolon is considered to be a comment for SDE and is not executed by the tool. You are encouraged to use more comments for clarity.

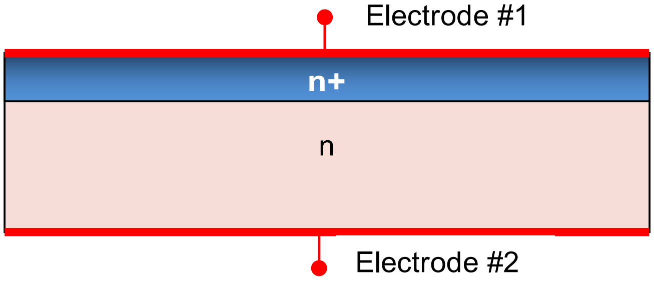


Figure SDE_01: Example used to explain commands of SDE

sde_dvs.cmd with description

```
(sde:clear)
```

Clear previous structure(s) and/or variable value(s)

```
(define tSub @tncSi@) ; um, thickness of the substrate
```

tncSi is a variable used to define thickness of n type cSi substrates. Value of tncSi defined in SWB is stored in tSub and this variable will be used for later operations. Same variable name, used in SWB, can be used for SDE. The advantage of doing this is that it is not necessary to use @ every time when the variable is used inside SDE.

```
(define node "@node@")
```

Saving current node number in variable named node as a string.

```
(sdegeo:set-default-boolean "ABA") ; new will replace old
```

Setting logical parameter stating the property of the overlapping region. “ABA” indicates that when a new region overlaps with an older region, the property (material) of the new region will be considered for the overlapping region.

```
(sdegeo:create-rectangle (position 0 0 0) (position 10 tSub 0) "Silicon" "Subs" )
```

Creating a rectangle named Subs using Silicon. Figure SDE_02 shows different aspects of the rectangle created by this command. Syntax to for that is

(sdegeo:create-rectangle (position x_start ystart z_start) (position x_end y_end z_end) material_name "region_name")

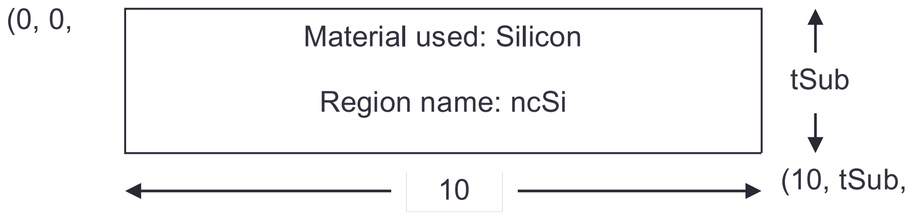


Figure SDE_02: Explanation of the command sdegeo:create-rectangle

```
(sdegeo:define-contact-set "electrode_1" 4  (color:rgb 1 0 0 ) "solid" )
```

Defining properties of contact set named electrode_1. Syntax to define a contact is the following

(sdegeo:define-contact-set name edge_Thickness edge_Color face_Pattern)

```
(sdegeo:define-2d-contact (find-edge-id (position 5 0 0) ) "electrode_1")
```

Placing the contact to an edge. For a 2D structure the contact should be placed at an edge and for a 3D structure the contact should be placed at a face. The command find-edge-id is used to find the edge id of a particular position.

```
(sdegeo:define-contact-set "electrode_2" 4  (color:rgb 0 1 0 ) "||" )
(sdegeo:define-2d-contact (find-edge-id (position 5 tSub 0) ) "electrode_2")
```

Defining and placing electrode_2 at the back. Commands are similar the ones used for electrode_1 except edge color for the contact, position of the contact and the face pattern are different.

```
(sdedr:define-constant-profile "Const.Subs"  "PhosphorusActiveConcentration" 4.4e15)
```

Defining constant doping profile named Const.Subs. A phosphorus doping of $4.4x10^{15}$ is used. `“BoronActiveConcentration”` can be used as species for p type doping definition. The syntax is the following

```
(sdedr:define-constant-profile definition-name species concentration)
```

```
(sdedr:define-constant-profile-region "PlaceCD.Subs"  "Const.Subs" "Subs")
```

Placing constant doping profile for the substrate region. The command is only valid to specify profile of a region. The syntax for the command is

```
(sdedr:define-constant-profile-region placement_name profile_name region_name)
```

```
(sdedr:define-refeval-window "window.nplusSi" "Line" (position 0 0 0) (position 10 0 0))
```

Defining reference/evaluation window to be used for Gaussian doping profile. For a 2D distribution a line is used as a window (Figure SDE_03). Direction normal to this line is used as a primary direction and directions along the line are considered for lateral distribution. Decay along lateral direction after the end of line can be defined by Gaussian or error function.

```
(sdedr:define-gaussian-profile "doping.profile.nplusSi"
"PhosphorusActiveConcentration" "PeakPos" 0 "PeakVal" 4.4e17 "StdDev" 0.15 "Gauss"
"Length" 0.1)
```

Defining Gaussian doping profile. Figure SDE_04 shows different parameters used to define Gaussian profile. The distribution in the primary direction can be defined in terms of length or standard deviation along primary direction. The effects of these parameters are shown by the formulas available in the diagram. Gaussian distribution is also selection for lateral distribution and a length of 0.1 is selected for lateral Gaussian distribution. One point should be noted that you are defining standard deviation for Gaussian profile and so if you want a particular junction depth you have to calculate the standard deviation.

For example, you want to dope an n-type substrate with doping concentration N_d with p-type dopant having Peak concentration, $N_{d\ peak}$, location of the peak concentration be 0, and a junction depth of d_L. Then at d_L the n and p type doping should be equal and you can calculate stdev using,

$N_d=N_{d\ peak}\ \exp[-0.5(d_L/Stdev)^2\ ]$

```
(sdedr:define-analytical-profile-placement   "place.nplusSi"   "doping.profile.nplusSi"
"profile.nplusSi" "Positive" "NoReplace")
```

Placing doping profile to the reference/evaluation window. The syntax for the command is

```
(sdedr:define-analytical-profile-placement     placement_name     doping_profile_name
refeval_window_name direction replace_option)
```

Direction value can be “Positive”/”Negative”/”Both”. The replace_option is to mention whether or not current doping will replace previous doping or just add a new doping on top of the previous one.

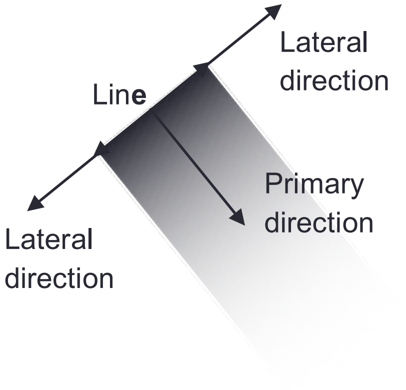


Figure SDE_03: Description of the window used for Gaussian doping profile

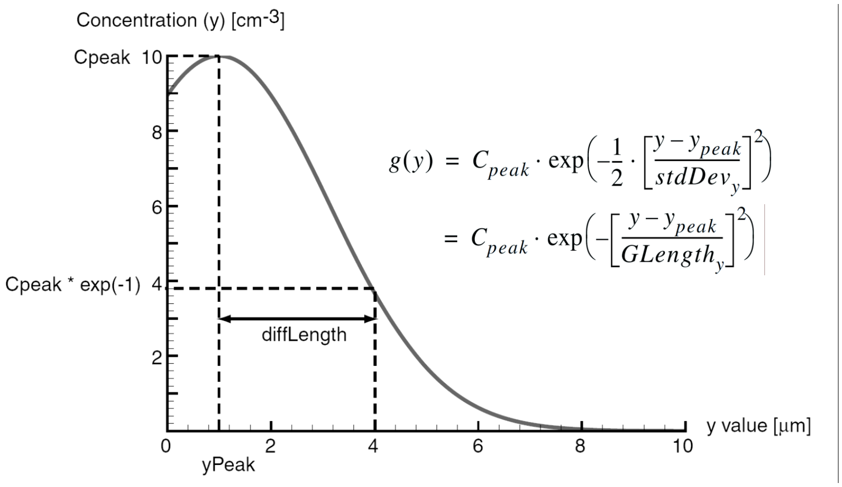


Figure SDE_04: General shape of Gaussian function

```
(sdedr:define-refeval-window  "refw.ncSi"  "Rectangle"  (position  0  0  0)  (position  10
tSub 0))
```

Choosing window for the grid placement. This comman is similar to window selection for Gaussian profile but here we are choosing a rectangular window.

```
(sdedr:define-multibox-size "refs.ncSi" 1 (/ tSub 5) 0 1 (/ tSub 20) 0 1 1.1 0)
```

Defining grid parameters. Multibox is especially helpful to define spatially varying grids. The syntax for the command is

(sdedr:define-multibox-size mbox_name max_x max_y [max_z] min_x min_y [min_z] ratio_x ratio_y [ratio_z])

Parameters for z axis is optional for 2D simulation.

```
(sdedr:define-multibox-placement "Place.ncSi" "refs.ncSi" "refw.ncSi")
```

Placing multi-box grid patternd. The syntax for the command is

(sdedr:define-multibox-placement placement_name size_definition_name window_name)

```
(sdedr:define-refeval-window "refw.nplus" "Rectangle" (position  0 0 0) (position  10
1 0))
(sdedr:define-multibox-size "refs.nplus" 1 (/ tSub 5) 0 1 (/ tSub 100) 0 1 1.2 0)
(sdedr:define-multibox-placement "Place.nplus" "refs.nplus" "refw.nplus")
```

Defining window, multibox size and placing multibox for highly doped region. It can be noticed that the window size is smaller along y-axis and minimum grid size along y (resolution along y) is also small.

```
(sde:build-mesh "snmesh" "" (string-append "n" node "_msh") )
```

Building the structure using snmesh mesh generation scheme. The structure can be used for other tools for simulation with all the doping and mesh parameters defined in sde_dvs.cmd file. The output files and all the log files will be named as nNODENUMBER_msh.extension. If node number 8 is the last column for SDE then the following files will be created.

`n8_msh.bnd,n8_msh.cmd, n8_msh.log and n8_msh.tdr`

And the structure represented by `n8_msh.tdr` is shown in Figure SDE_05.

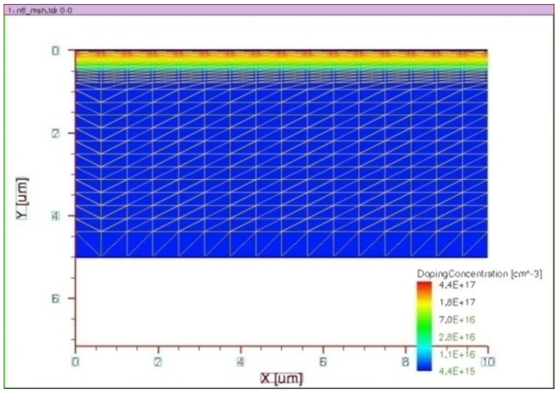


Figure SDE_05: Structure for Project 0 with doping and mesh defined by the sde_dvs.cmd

To View the Structure

Once structure is successfully created you can view the structure after following steps.

Right click on the cell of the last column under SDE, select Quick Visualize.

Debugging the code:

If there is a major error then the log window that will appear after running SDE will end with a message saying ``gsub exits with status 1``. In that case read the error message on the pop-up/log window or the file named glog.txt in you project folder. If the status is 0 then check the log file named n(NodeNumber)_dvs.cmd.log [for Project_0 it is n8_dvs.com.log]

Make sure there is no error displayed in the log file. If there is any error then look for the corresponding command line of sde_dvs.cmd and fix it. All lines of sde_dvs.cmd should appear in the log file with the status after running the command line. If somehow you don't see sections after a particular line then there must be a problem with the first missing line.

Sentaurus Device (SDevice)

Sentaurus device is the main tool that takes structure of a device and its material parameters and runs the simulation for different input parameter variation. As mentioned earlier sdevice_des.cmd is the file that has to be edited to change different aspects of a simulation. The sdevice_des.cmd for the project zero is used in the following section to describe different command in details.

sdevice_des.cmd with description

```
#setdep @previous@
```

Setting dependence on previous node(s). So, you can't run this node unless the structure is created already.

```
File {
*-Input
      Grid     = "@tdr@"
      Parameter  = "@parameter@"
      * Output Files
      Current    = "@plot@"
      Plot       = "@tdrdat@"
      Output     = "@log@"
}
```

Defining input and output file for the tool. tdr files created by previous node will be used for input grid file. Files with extension .par, in our case sdevice.par will be used to define material properties. Current voltage variation will be saved in .plt file with current node number (last node of sdevice) mentioned in the file name. Different spatial parameters like electron/hole distribution will be save in file with .tdr extension. And final log while running sdevice will be saved in file with .log extension. In our case file for Current, Plot and Output are n95_des.plt, n95_des.tdr and n95_des.log, respectively.

```
Electrode {
   { Name="electrode_1" Voltage=0.0 }
   { Name="electrode_2" Voltage=0.0 }
}
```

Defining initial properties of electrode. More detailed syntax where other properties can also be defined is the following

```
{Name= electrode_name Voltage= init_voltage Barrier= barrier_value AreaFactor=
area_factor_value}
```

where Barrier is the metal–semiconductor work function difference or barrier value for an electrode. Area factor is useful to represent actual dimension of the electrode in z-direction for a 2D simulation. By default it is one micrometer in z-direction. So, if you consider something different that area factor can be changed properly to have representative current values. Area factor is multiplied with the current (I) value obtained from the simulation.

```
Physics {
   Temperature=@temp@
   EffectiveIntrinsicDensity( Slotboom )
   Mobility ( DopingDep eHighFieldSaturation hHighFieldSaturation )
   Recombination ( Auger SRH(DopingDep) )
}
```

Defining the physics that has to be considered. Temperature is set to the temperature define in SWB. The temperature dependent intrinsic density is considered and Slotboom model is used. Mobility is considered to be doping and electric field dependent. Doping dependent Shockley-Reed-Hall recomobination and Auger recombination is considered. Material parameter file consists all the model related parameters. For now you can skip details of the physics section.

```
Insert = "PlotSection_des.cmd"
Insert = "MathSection_des.cmd"
```

Inserting command file that describes what 2D distribution profiles (like eDensity) should be saved and what mathematical parameters should be selected to simulate the structure. You don’t have to change these parameters and can skip details of these command file.

```
Solve{
  NewCurrentFile="init"
  Poisson
```

Solving Poisson’s equation with initial condition that is the two electrodes are at voltage equal to zero.

```
  NewCurrentFile=""
  Coupled (Iterations=100) { poisson electron hole  }
```

Solving Poisson’s equation along with drift-diffusion model for electron and hole for initial condition. It will look for a convergence within 100 attempt.

```
  Quasistationary (
      Goal { Name="electrode_1" Voltage=5 }
      InitialStep=4e-2 Increment=1.1 Minstep=1e-3 MaxStep=1e-1
      ) {Coupled (Iterations=100) {Poisson Electron Hole}}
```

It is important to understand this command. The Quasistationary statement is to look for a series of quasistatic or steady state ‘equilibrium’ solutions where for a particular instance the voltage(s) or current(s) of electrode(s) is considered to be constant or steady. A Goal or set of Goals for one or more electrodes are defined. For this example, goal is to set the voltage of electrode_1 to 5 V. Multiple Quasi-stationary commands can be used to reach a goal using different step sizes.

Details of how to reach the goal are mentioned in the next line. The constraints on the step size are proportional to the normalized Goal (t=1). So, initial step is 5 V times 4x10^{-2} that is, 2x10^{-1} V. It is assumed that convergence is achieved at each step. If, at any step, there is a failure to converge, Sentaurus Device performs automatic step size reduction until convergence is again achieved, and then continues the simulation.

```
      CurrentPlot ( Time = (Range = (0.0 0.2) Intervals=10; Range = (0.2 1.0)
Intervals=20))
```

CurrentPlot is to save which data points of voltage or current variation to be stored in CurrentPlot file. Even though the simulation is carrier out for smaller steps different step size can be used to store the simulation result. The total range of quasi-stationary will be the reference and the range mentioned in the command is relative to the total range. In this command there will be 10 intevals for 0 to 1 V and 20 intervals for 1 V to 5 V electrode_1 voltage variation.

```
      System("rm -f init*") *remove the plot we don’t need anymore.
}
```

To View Details of the Simulation

n95_des.log and n95_des.out files of the project folder is used to store details of the simulation result. So, these two files can be opened to view the details of simulation even when the code is running. This approach is especially helpful if it takes long time to simulate a structure. If there is an error, these two files along with n95_des.err can be used to figure out the problem.

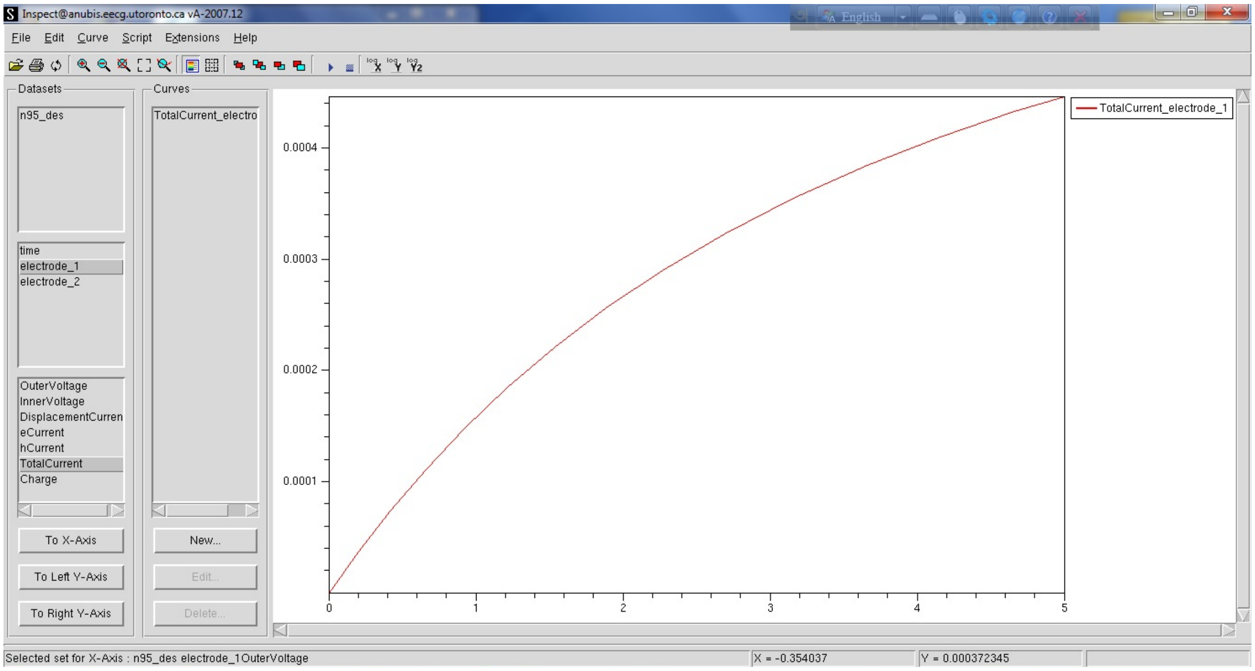
Inspect

Inspect can be used to plot voltage or current variation utilizing the .plt file created after simulating a cell structure. Command line written in inspect_ins.cmd can be executed from SWB interface. Moreover, Inspect can be directly opened. The parameters to draw then can be selected after loading the project.

Working with Inspect Program

Type inspect at the terminal to load the software and press enter.

Interface



Selected set for X-Axis : n95_des electrode_1OuterVoltage

X = -0.354037

Y = 0.000372345

Loading a Dataset

File > Load DataSet… (Ctrl+L) > Go to the Project Directory > select *.plt file

Plotting a curve

Once a project is loaded choose parameters for x and y axes. To plot outer voltage of electrode_1 verses Total current of electrode_1 take following steps:

1. Select electrode_1, select OuterVoltage, click ‘To X-Axis’
2. Select electrode_1, select TotalCurrent, click ‘To Left Y-Axis’

Changing Plot Properties

Edit > Axes > Choose properties you want to set

Saving the image as eps

File > Write EPS > Choose file location and file name and save

Exporting Data in a csv file

File > Export > CSV (Comma Delimited) > Choose file location and file name and save

Working with Inspect from SWB

Command lines written in inspect_ins.cmd can be executed from SWB by running the node under Inspect tool. Command descriptions are provided for the command used in the file for Project 0.

Commands for Inspect

```
#setdep @previous@
```

Setting dependency on previous node.

```
set N @node@
```

Setting value of the variable N.

```
proj_load  @plot@ CURVE_NAME($N)
```

Loading project. @plot@ will load the .plt file available in the folder and project name is CURVE_NAME(95) if 95 is the value of current node.

```
cv_createDS  IV($N)  "CURVE_NAME($N)  electrode_1  OuterVoltage"  "CURVE_NAME($N) electrode_2 TotalCurrent" y
```

Creating the curve and displaying the curve. The syntax of the command is

`cv_createFromScript curveName xdata ydata {axis y}`

where,
`curveName`, a unique name for the new curve
`xdata`, a list of data to use for the x-dataset
`ydata`, a list of data to use for the y-dataset
`axis`, optional parameter specifying the axis to use; the default is y; the options are y or y2

```
cv_setCurveAttr IV($N)  "IV" blue solid 2 circle 0 defcolor 1 defcolor cv_abs IV($N) x
```

Setting the curve attribute. The syntax for this command is

`cv_setCurveAttr curveName legend color style width shape size outColor outWidth fillColor`

where,
`curveName`, the curve name
`legend`, the curve legend
`color`, the color of the curve line
`style`, the drawing style of the curve line (`solid, dashed, dotted, "long dashed", or "long dotted"`)
`width`, the width of the curve line
`shape`, a keyword for the marker shape (`none, square, circle, diamond, plus, cross, splus, scross, or triangle`)

`size`*, the marker size*
`outColor`*, the color of the marker outline*
`outWidth`*, the width of the marker outline*
`fillColor`*, the fill-in color of the marker*

```
gr_setAxisAttr "X" "Voltage (V)" 20 0 5 black 1 15 0 1 lin
gr_setAxisAttr "Y" "Current (A)" 20 -5e-4 0 black 1 15 0 1 lin
```

Setting the axes attributes. The syntax for this command is

```
gr_setAxisAttr axis title tfont min max color width font angle div scale {tcolor}
```

where,

`axis`*, a keyword (X, Y, or Y2) specifying an axis*
`title`*, the axis title*
`tfont`*, the font size of the axis title*
`min, max`*, minimal and maximal values of the axis*
`color`*, the color of the axis*
`width`*, the width of the axis line*
`font`*, the font size of the tick label*
`angle`*, the angle at which the tick labels are drawn*
`div`*, the number of secondary ticks between the main ticks*
`scale`*, specifies linear (lin) or logarithmic (log) display of the axis*
`tcolor`*, the color of the axis title*

Sentaurus Visual

Sentaurus Visual (SVisual) can be used to plot 2D and 3D variation utilizing the .tdr file created after simulating a cell structure. SVisual can be directly opened by typing svisual in the terminal window or in Sentaurus Workbench going to Extensions > Run Sentaurus Visual. The parameters to draw then can be selected after loading the project.

Loading a Project

To load a *.tdr file after a successful simulation click File>Load>select proper tdr file. As mentioned before after running Sentaurus Device, results will be saved in a file having n and node number at the beginning of a filename.

Sidebar and Top Menu

Several operations for SVisual are available in the sidebar. Most are in the top menu.

Precision Cuts

This is useful for taking a cross-section of a 3D device or to see variation of a value along a particular axis. In the menu, choose Tools > Precision Cut. Then you can define the cuts you wish to make, whether along a specific axis (Orthogonal) or an arbitrary direction (Free). You can also create multiple cuts.

This is should be enough to be introduced with Sentaurus. There are also lots of resources on the web should you want to look up more details.

Have fun!

RANDOM STUFF – DO NOT INCLUDE

Command To Open Different Tools

Once the path is set then you can open different tool from Terminal.

Sentaurus Workbench

1. Type *swb* in the terminal window
2. Choose your home directory when the window appears.

SVisual

Type *svisual* in the terminal window

Inspect

Type *inspect* in the terminal window

Sentaurus Structure Editor

To open the SDE tool type *sde* in the terminal window.

[Normally you need to edit the sde_dvs.cmd file and you can do it by opening the file from SWB. Only to run individual or multiple command used in the sde_dvs.cmd file you can open the tool. We recommend use of this tool to advanced users.

Note: You can open different tools within Sentaurus by going under the “Extensions” in the Sentaurus Workbench and choosing to run the various tools (like Structure Editor, Process, etc.)