# Experiment 1

## Introduction to Linux and Sentaurus TCAD Environment for semiconductor device and process simulation

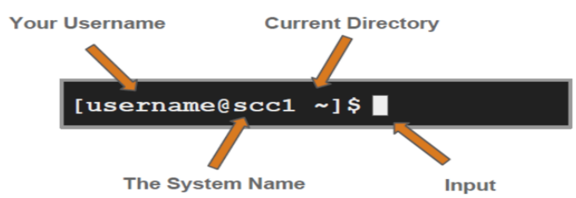
**Objective:** To familiarize learners with the Linux operating system and its command-line interface, and to provide hands-on experience with the Sentaurus TCAD simulation environment for performing device-level simulations of advanced transistors, including structure creation, process simulation, physical modeling, and analysis of electrical characteristics.

**Theory:** Linux is a family of open-source Unix-like operating systems based on the Linux kernel assembled under the model of free and open-source software development and distribution. The operating system (OS) is the software that directly manages a system's hardware and resources, like CPU, memory, and storage (RAM, RAM and other memory devices). Comes in several "distributions" to serve different purposes.

### Why Linux

- Free and open source.
- Powerful for research data centers
- Personal for desktops and phones
- Universal
- Community (and business) driven.

Linux: "prompt"




### Some basic commands are:

- ls: List directory contents.
- cd: Change the current directory.
- pwd: Print the working directory.
- mkdir: Create directories.
- rm, rmdir: Remove files and directories.
- cp, mv: Copy and move files or directories.

## TCAD and the Semiconductor Industry

Technology Computer-Aided Design (TCAD) refers to using computer simulations to develop and optimize semiconductor processing technologies and devices. TCAD simulation tools solve fundamental, physical, partial differential equations, such as diffusion and transport equations, for discretized geometries, representing the silicon wafer or the layer system in a semiconductor device. This intensive physical approach gives TCAD simulation predictive accuracy.

Therefore, it is possible to substitute TCAD computer simulations for costly and time-consuming test wafer runs when developing and characterizing a new semiconductor device or technology.

TCAD simulations are used widely in the semiconductor industry. As technologies become more complex, the semiconductor industry relies increasingly more on TCAD to cut costs and speed up the research and development process. In addition, semiconductor manufacturers use TCAD for yield analysis, that is, monitoring, analyzing, and optimizing their IC process flows, as well as analyzing the impact of IC process variation.

## Process Simulation

In process simulation, processing steps such as etching, deposition, ion implantation, thermal annealing, and oxidation are simulated based on physical equations, which govern the respective processing steps. The simulated part of the silicon wafer is discretized (meshed) and represented as a finite-element structure (Figure 1).

## Device Simulation

Device simulations can be thought of as virtual measurements of the electrical behavior of a semiconductor device, such as a transistor or diode. The device is represented as a meshed structure. Each node of the device has properties associated with it, such as material type and doping concentration. For each node, the carrier concentration, the current densities, the electric field, the generation and recombination rates, and so on are computed (Figure 2).

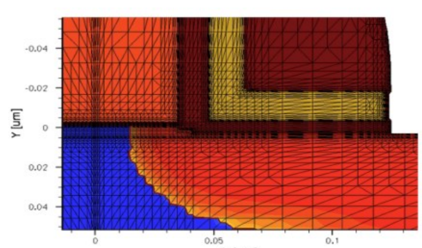



*Figure 1. Magnification of gate-drain corner of an NMOSFET with finite-element grid. (Click image for full-size view.)*

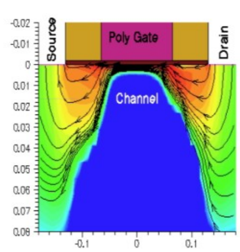



*Figure 2. Current flow lines in a 0.13 µm NMOSFET at $V_{gs}$ = 1.5 V and $V_{ds}$ = 3.0 V; shading represents current density.*

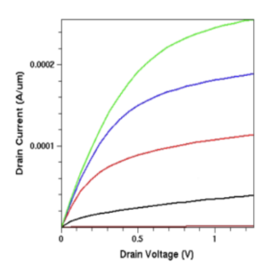
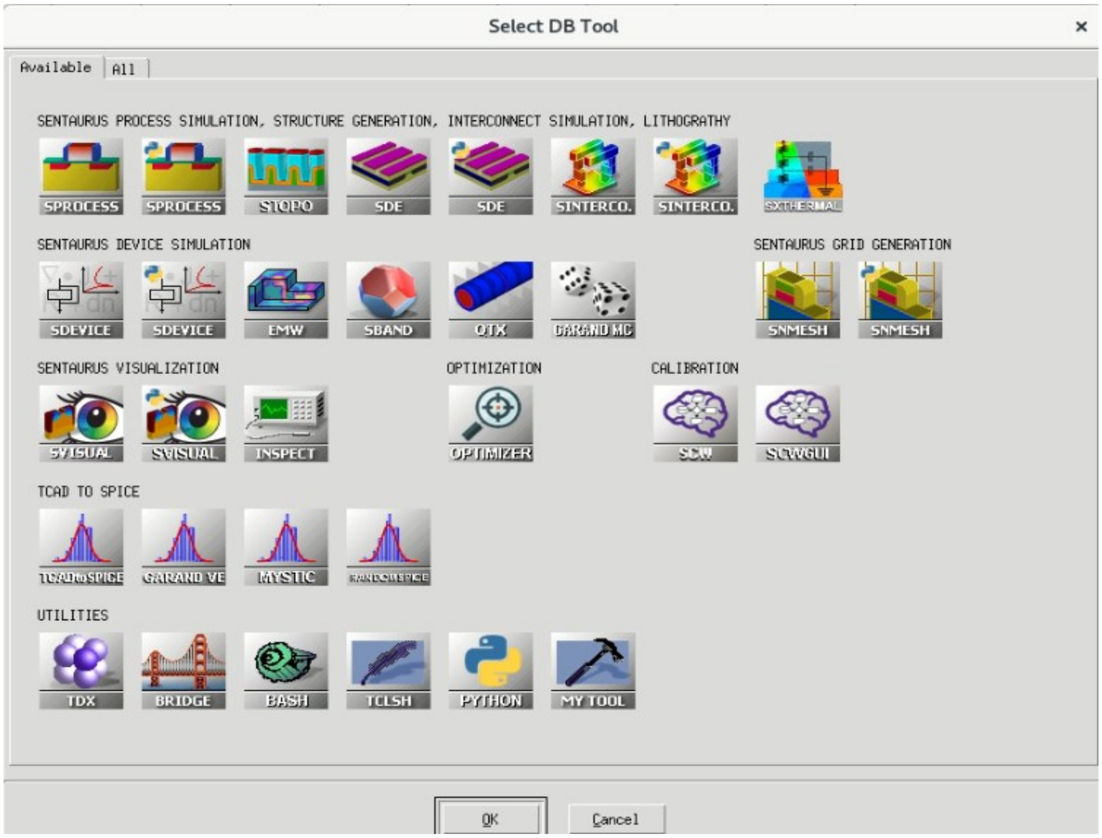



*Figure 3. Drain current as a function of drain voltage for a 50 nm NMOSFET at $V_{gs}$ = 0.25, 0.5, 0.75, 1.0, and 1.25 V.*

**List of tools available with Sentaurus TCAD**




The various tools are available with Sentaurus TCAD, but in this lab, we will focus on the following tools

**a. SWB**

A key component of this workflow is the Sentaurus Workbench (SWB), an integrated graphical user interface that serves as the central platform for managing simulation projects. SWB allows users to organize and execute simulations through a visual flow diagram that links different stages—structure generation, meshing, physical simulation, and post-processing—into a unified environment. It simplifies the simulation setup by providing templates, drag-and-drop functionality, parameter control, and automated job execution, making it highly efficient for both beginners and advanced users. Through SWB, users can also manage multiple simulation variants, perform parametric sweeps, and analyze results, thereby streamlining the entire TCAD simulation workflow.

**b. Sentaurus structure editor (SDE)**

Sentaurus Structure Editor is a graphical tool used for creating and visualizing 2D/3D semiconductor device geometries. It allows users to define layers, regions, electrodes, and doping profiles in an intuitive GUI environment. SDE is especially useful for complex custom structures that are difficult to define using scripts.

**c. Sentaurus Process (Sprocess)**

Sprocess is a process simulation tool that models the physical fabrication steps of semiconductor devices, such as ion implantation, diffusion, oxidation, deposition, and etching.

**d. SNMESH (snmesh)**

SNMESH is a meshing tool used to convert the geometric structure into a numerical mesh suitable for simulation.

**e. Sentaurus device (Sdevice)**

Sdevice is the core physical simulator in Sentaurus TCAD that solves semiconductor transport equations to model the behavior of electronic and optoelectronic devices.

**f. SVISUAL Tool (Svisual)**

Svisual is a visualization tool used to inspect the structure and simulation results graphically, such as doping distribution, potential, current flow, or carrier densities.

**g. INSECT (inspect)**

Inspect is a command-line-based visualization and plotting tool used for extracting and analyzing simulation results like I–V characteristics, capacitance, and transient responses.

**TCAD Simulation Flow**

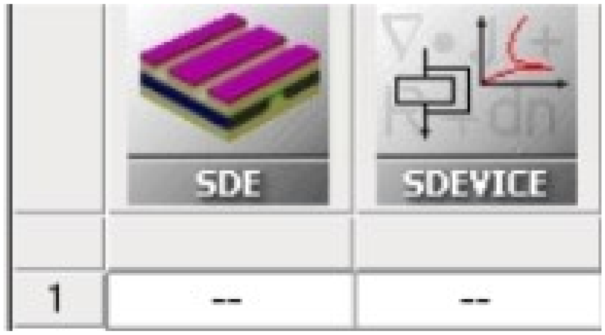
A typical TCAD simulation flow is integrated in a Sentaurus Workbench project, and the flow usually consists of two main steps (see the Sentaurus Workbench module):

- **Structure generation**
- **Device simulation**

**Two different approaches can be used for the structure generation:**

- **Process simulation** performed by Sentaurus Process is based on physical models for each step that is performed to fabricate a device


- **Process emulation or geometric definition** of the device, performed by Sentaurus Structure Editor (see the Sentaurus Structure Editor module). It uses 2D or 3D predefined geometric objects in combination with analytically defined doping profiles and profiles loaded from the process simulation step.




**Applications**

- Advanced CMOS device simulation (FinFET, Nanosheet, Forksheet FETs)
- Study of short channel effects (SCEs), mobility degradation, DIBL, and subthreshold slope
- Thermal analysis including self-heating and interfacial thermal resistance (HITR)
- Ferroelectric NCFET simulation using Landau parameters
- Reliability analysis (e.g., Hot carrier effects, Bias Temperature Instability)
- Device-circuit co-simulation via mixed-mode operation

## Usage and Workflow Overview

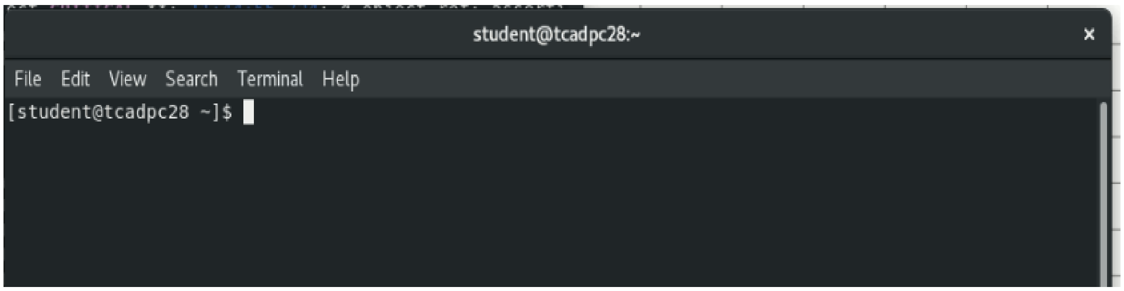
1. Environment Setup
   - Use a Linux terminal environment to load the Synopsys setup files (e.g., via the `source` command)
   - Launch Sentaurus Workbench or run components via the command line
2. Project Creation
   - Define a simulation project in Sentaurus Workbench
   - Create or import structure files using tools like Sentaurus Structure Editor (SDE) or Structure Generator (SDEGEO)
3. Meshing
   - Use Sentaurus Mesh to discretize the geometry for numerical solving
4. Device Simulation
   - Define physical models, materials, doping profiles, boundary conditions, and biasing schemes in Sentaurus Device
   - Run simulations to analyze electrical output characteristics (e.g., ID-VG, ID-VD curves), transient response, or thermal behavior
5. Post-Processing
   - Use Inspect or Tecplot to visualize simulation results (current flow, temperature, band diagram, etc.)
6. Automation & Scripting
   - Batch simulations, parameter sweeps, or statistical variation analysis using shell or TCL scripts

**How to open the Tools**

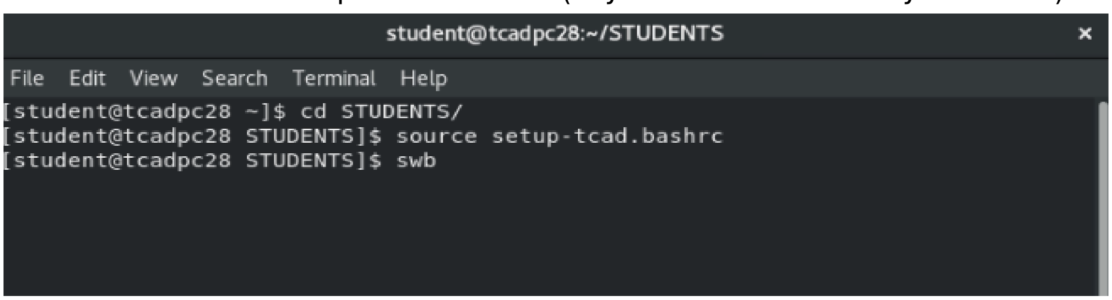
This section is divided into various sections, and each section will be discussed separately.

**Section A: Access to SWB tools**

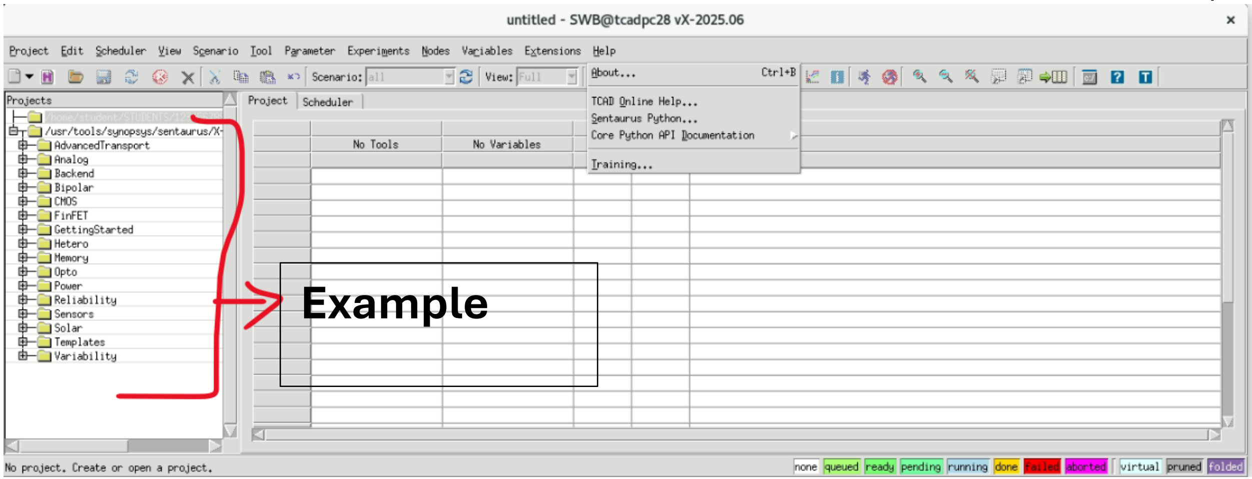
1. Create a folder with your registration number (only once and without any spaces) inside the path: `Home > STUDENTS`.
2. Copy tcad.bashrc file into your folder, which you created with your registration number (Do this only once)
3. Open the terminal (from your folder by right click )



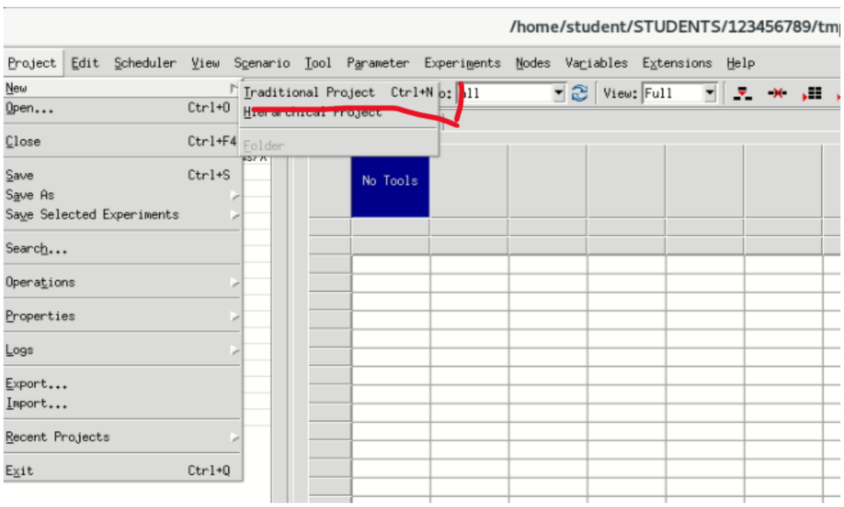

4. Write this command:
   a. cd STUDENTS
   b. source setup-tcad.bashrc
   c. swb/sde/sprocess/Svisual (any of the one based on your needs)



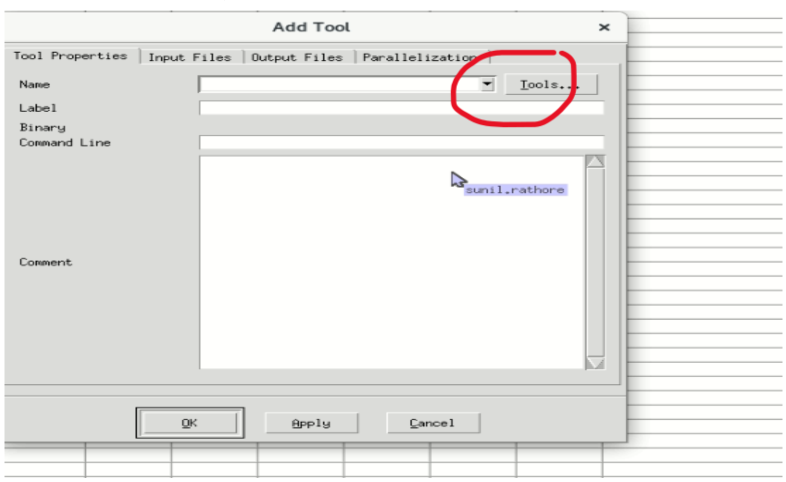

**5.** Now use the swb command to open the Sentaurus workbench in the command line (mentioned in step 3)
   a. From help training and manual can be opened
   b. Below right side of the color code shows the working status of the nodes.



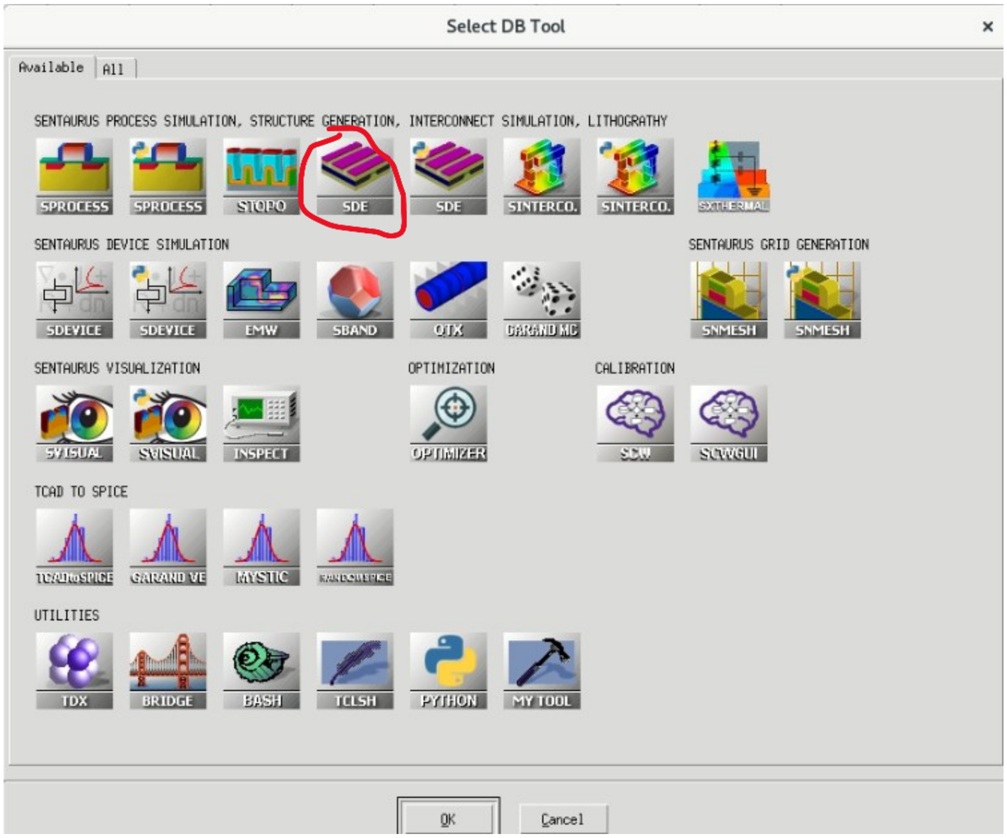

6. Create new project > New > Traditional




7. Right-click on No Tools, then Add




8. Click on Tools
   The list of available tools will be visible for the usage



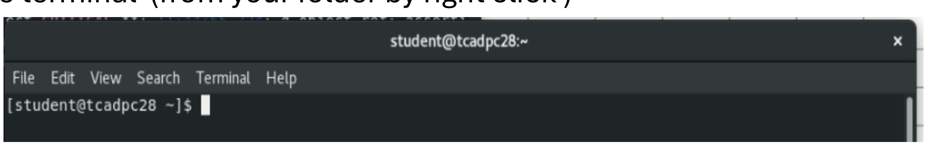

9. Click on SDE (4th place tool), then OK, then again OK

   SDE (Sentaurus Editor tools will be added in the SWB

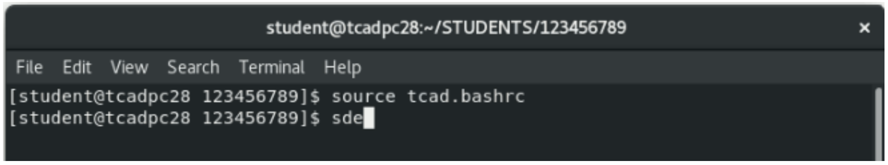
10. Similarly, by clicking on the previous tools (in this case, SDE), the next tools can be added

## Section B: Access to SDE tools and create a structure

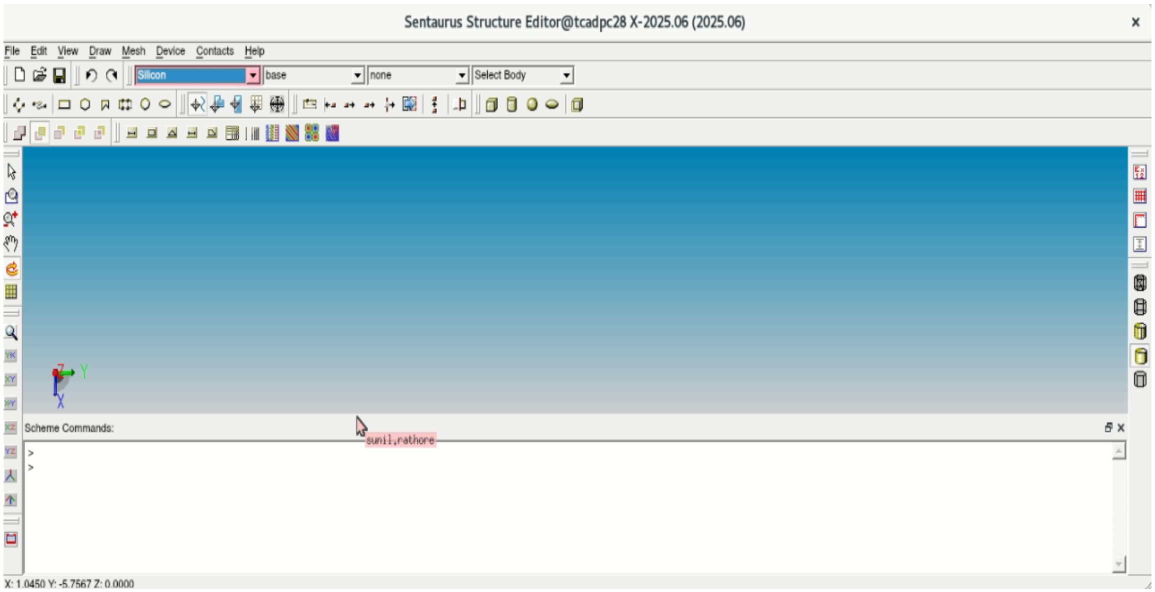
1. Clean
2. Create a folder with your registration number (only once and without any spaces) inside the path: `Home > STUDENTS`.
3. Copy tcad.bashrc file into your folder, which you created with your registration number (Do this only once)
4. Open the terminal (from your folder by right click )




5. Write this command:
   a. cd STUDENTS
   b. source setup-tcad.bashrc
   c. sde




6. The GUI of the Sentaurus Structure Editor will open



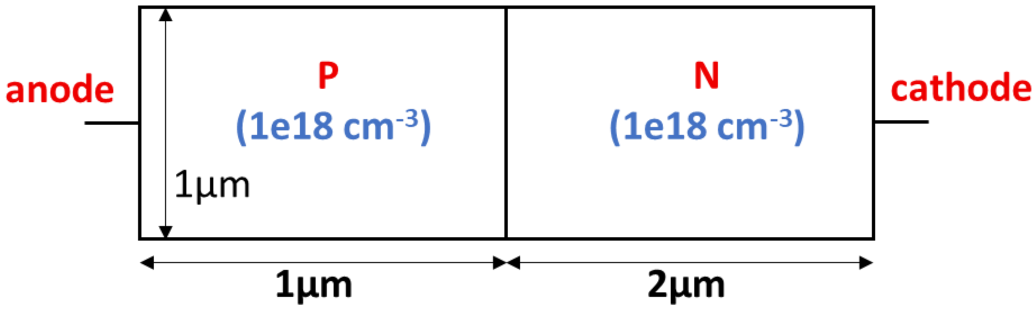

7. There are various options available in **Manu Bar** as listed below

| Menu | Description |
|---|---|
| File | Load, save, and print functions |
| Edit | Change existing geometric objects |
| View | Visualization preferences and auxiliary views |
| Draw | Drawing and basic object creation, preferences |
| Mesh | Define a meshing strategy, call the meshing engine, visualize the generated mesh and data fields |
| Device | Define doping profiles |
| Contacts | Define and edit contacts and contact sets |
| Help | Version information |

8. Input and Output File Types
    a. Scheme script file (.scm)
    b. Journal file (.jrl)
    c. ACIS SAT file (.sat)
    d. TDR boundary file (_bnd.tdr)
    e. Doping and refinement file (.cmd)
9. To save the file

    File > Save Model as > save all format file as mentioned in point 8.

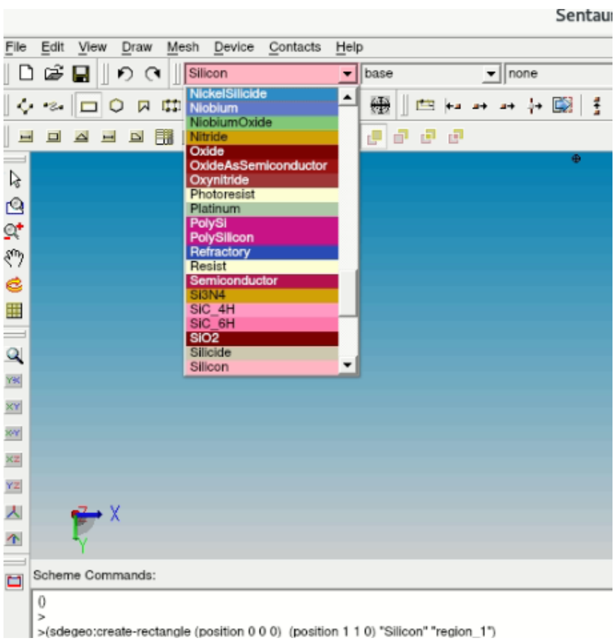
**10. To create a structure given below consisting of doped silicon on both sides with P and N-type doping with a doping concentration of 1e18.**




a. Reinitialize Sentaurus Structure Editor by choosing File > New.

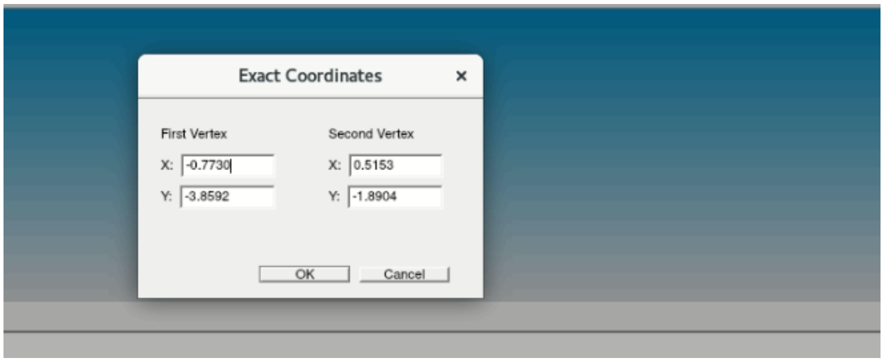
b. This step is always recommended before creating a new model.

c. From the Material list, select Silicon.

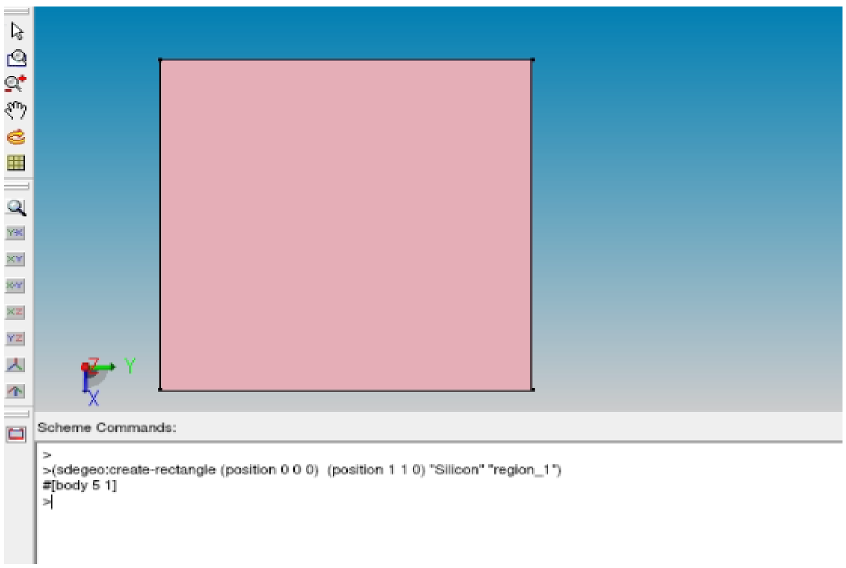



d. Click the Isometric View (Isometric View button) button to change to the 2D isometric view.

11. Choose Draw > Exact Coordinates to switch on the Exact Coordinates mode.
12. Choose Draw > 2D Create Tools > Create Rectangular Region.
13. Choose Draw > Untick Auto Region Naming to switch on the exact naming of the region.
14. Draw the shape by dragging the pointer in the view window.
15. When the mouse button is released, the Rectangular Coordinates dialog box opens, showing the coordinates of the drawn shape.



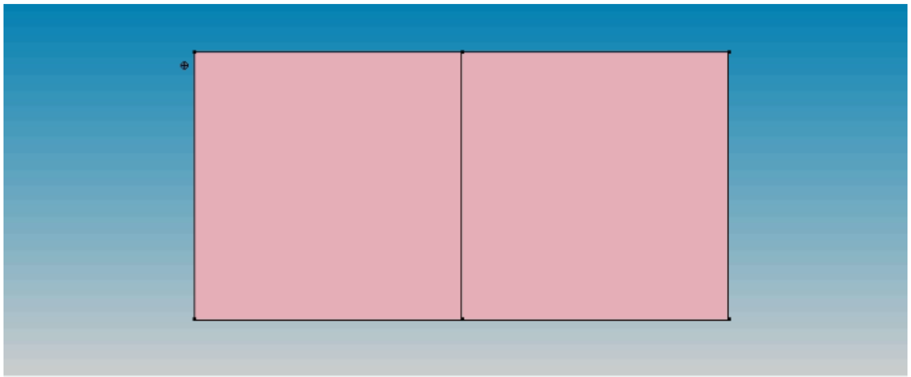

16. Enter (0 0) and (1 1) for the start and end corners of the box.
17. Click OK to create the rectangular box.




18. The last Scheme script command appears in the command-line window:
    (sdegeo:create-rectangle (position 0 0 0) (position 1 1 0) "Silicon" "n_region")

    To access the command-line window, click in the area. All previous commands printed in this window can be edited and reused. To illustrate this, edit the last command so that it becomes:
19. Another rectangular region is created for the p_region with the same steps mentioned above

(sdegeo:create-rectangle (position 1 0 0) (position 2 1 0) "Silicon" "p_region")

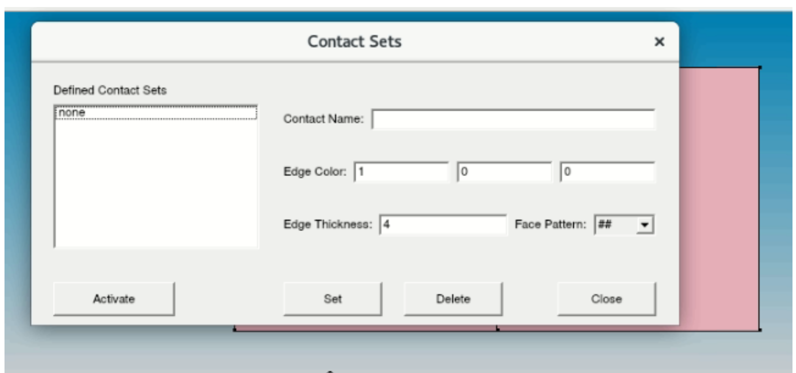


20. Defining Contacts

Contacts can be defined to allow the constructed device to be connected to outside power sources.

**To define a contact:**

a. Choose Contacts > Contact Sets.



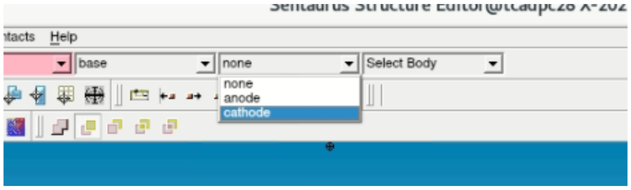

b. The Contact Sets dialog box opens.
c. In the Contact Name field, enter the name of the contact.
d. In the Edge Color fields, specify RGB colors.
e. In the Edge Thickness field, modify the value to mark the contact.
f. The Face Pattern field is effective only for marking 3D contacts.
g. Click Set.
h. The contact is added to the Defined Contact Sets pane.
i. Define additional contacts as required.
j. When all the contacts have been defined, click Close.
k. **Corresponding command:**

(sdegeo:define-contact-set "anode" 4  (color:rgb 1 0 0 ) "##")

(sdegeo:define-contact-set "cathode" 4  (color:rgb 1 1 0 ) "##")

21. Setting Contacts at Existing Edges

a. Choose Contacts > Contact Sets.




b. The Contact Sets dialog box opens.

c. Select the required contact from the Defined Contact Sets pane, for example, source.

d. Click Activate to activate the selected contact.

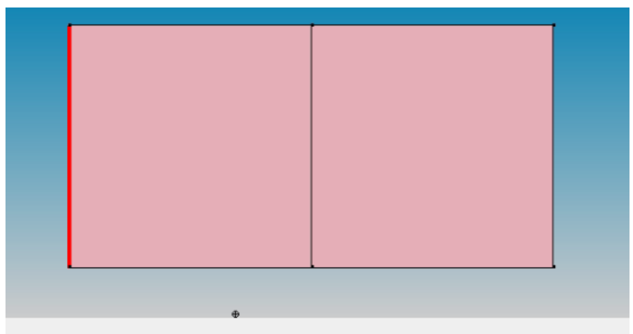
e. Alternatively, a contact can be activated from the Contact list.

f. In the Selection Level list, choose Select Edge to ensure that only edges of the device can be selected in the next steps.

g. Click the Select (Select button) toolbar button.

h. Click the edge of the structure where the contact, for example, anode, will be defined. The selected edge is now highlighted.

i. Choose Contacts > Set Edges.



j. This defines the selected contact, for example, Anode, at the highlighted edge. The edge is now characterized by the color and line styles previously set for the contact.

k. Repeat the above operations to associate the cathode contact with the silicon epilayer edge to the right of the PN Diode.

l. **Corresponding command line:**

```
(sdegeo:set-current-contact-set "anode")
(sdegeo:set-contact (list (car (find-edge-id (position 0 0.5 0)))) "anode")
(sdegeo:set-current-contact-set "cathode")
(sdegeo:set-contact (list (car (find-edge-id (position 2 0.5 0)))) "cathode")
```

22. Saving the Model

To save a model, choose File > Save Model, or press Ctrl+S, or click the corresponding toolbar button.

23. Generating Doping Profiles

To introduce the most important 1D/2D doping definitions of Sentaurus Structure Editor. There are various types of doping profiles that are possible, such as

   a. Constant doping profiles
   b. Analytical/Gaussian doping profiles

24. In the PN diode, we are going to do the Constant doping profiles
   a. In the P side, the boron doping of $1e^{18}$ $cm^{-3}$
      1. Go to Device> Constant Profile Placement> Click
      2. Placement name> ConstantProfilePlacemdnt_P
      3. Region>P
      4. Constant Profile Definition> ConstantProfileDefinition_P
      5. Species> BoronActiveConcentration
      6. Conceration>1e18
      7. NoRepace
      8. Add Placement

*(sdedr:define-constant-profile "ConstantprofileDefination_P" "BoronActiveConcentration" 1e18)*
*(sdedr:define-constant-profile-region "ConstantProfilePlacemdnt_p" "ConstantProfileDefinition_P"* "P_region")

   b. In the N side, the phosphorus doping of $1e^{18}$ $cm^{-3}$

   Repeat the above steps for the N-type region

      1. Go to Device> Constant Profile Placement> Click
      2. Placement name> ConstantProfilePlacemdnt_N
      3. Region>N
      4. Constant Profile Definition> ConstantProfileDefinition_N
      5. Species> PhosphorusActiveConcentration
      6. Conceration>1e18
      7. NoRepace
      8. Add Placement

*(sdedr:define-constant-profile "ConstantprofileDefination_N" "PhosphorusActiveConcentration" 1e18)*
*(sdedr:define-constant-profile-region "ConstantProfilePlacemdnt_N" "ConstantProfileDefinition_N"* "N_region")

25. Define the Mashing
   1. First, define the Mashing refinement window
      a. Go to Mash>Define Rel/Eval Window> Rectangular
      b. Select all areas using drag
      c. Assign the coordinate (-0.5, 1.5) (2.5, -0.5 )
      d. Click OK

2. Secondly, refinement placement
    a. Go to Mash, Refinement Placement> Refinement Specification Box open
    b. Change the Placement Name: RefinementPlacement_Global (As mashing defines globally or regionally)
    c. Select the Ref/Eval Window as Global
    d. Refinement Definition > RefinementDefinition_Global
    e. Define the size of Max and Min Elements
    f. More>> Interface Length> tick on UseRegionNames
    g. In Refinement Functions> Select Interface Length > Select p_region and n_region > Value 1> Factor 0.8 > add
    h. Click on Create Refinement > Close
    i. The global mashing is defined

    Code:

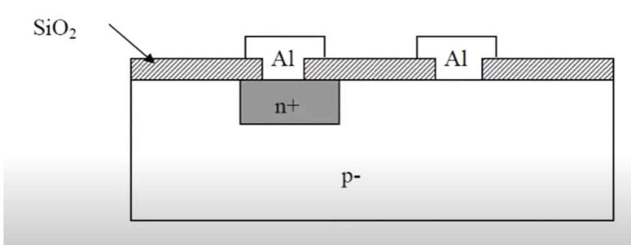
26. File> Save Model >
27. Mash > Build Mesh > give the appropriate file name > build Mesh
28. Svisual Open
29. Go to folder> open jrl file (which you saved at the beginning of creating the structure)> read the code. > Clean the file

We are going to add this code to the sde of swb

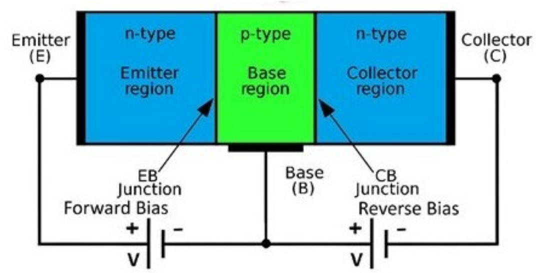
1. Open SWB as mentioned above add the sde tool
2. Right click on sde> edit command> open the gedit file
3. Paste the clean jrl file and save it
4. Run the simulation tool

**Exercise:**

1. Create the structure given below using the code/GUI method.
2. Create and visualize the profile of the given structure below




3. Create and visualize the profile of the given structure below

4. 
Emitter
(E)
n-type
Emitter
region
p-type
Base
region
n-type
Collector
region
Collector
(C)
EB
Junction
Forward Bias
Base
(B)
CB
Junction
Reverse Bias
+
-
V
+
-
V