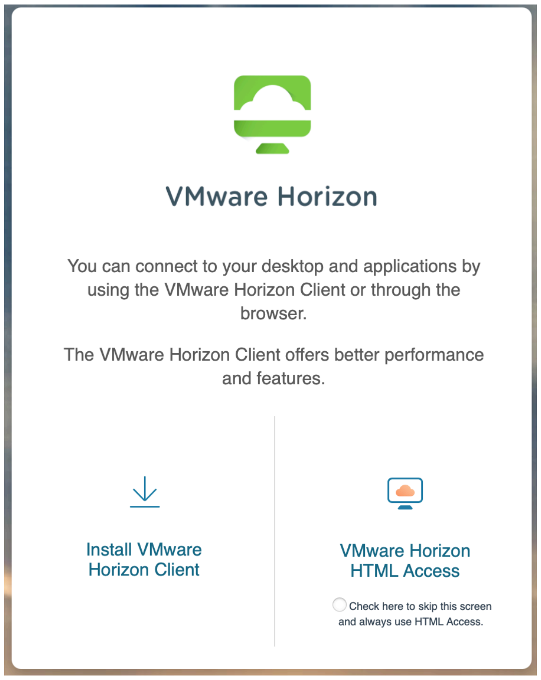
Quick guide to using Silvaco SUPREM

Getting started

The Silvaco software runs on a bank of Windows computers that can be accessed remotely. There are four licenses, so four different students can use SUPREM simultaneously. (Consequently, there may be instances where we will not be able get in, if all the available slots are being used.)

1. Go the web site: vlab.ece.iastate.edu.



VMware Horizon

You can connect to your desktop and applications by using the VMware Horizon Client or through the browser.

The VMware Horizon Client offers better performance and features.

Install VMware Horizon Client
VMware Horizon HTML Access
Check here to skip this screen and always use HTML Access.

2. We can choose to download the VMWare Horizon Client software or to run from inside a browser window. In general, using the client software directly on a personal computer sitting in front of us will probably give better performance, but that requires us to download and install the client. To use the client, click on the "Install…" option above, choose and download the correct software for our computer, install it, and then launch the "VMWare Horizon Client" app on the computer. To use the web browser option, then click on the right icon to be taken to a browser window to start the process. Once we have chosen the method for access, then everything that follows is the same.

3. Log in with your ISU net-ID and password.

4. After login, we should see a screen with four icons representing various options for EE and CprE class software. The SUPREM software is in the "EE" group. Double-click on the EE icon and wait patiently for Windows to launch.

5. On the Windows desktop are several short-cut icons, including one for "S.EDA Tools". Double-click to start the Silvaco software.

6. In the file navigation window that opens, double-click on "Deckbuild". (It may take a few seconds to launch — the license must be the checked and of course everything is running

over a network.) The program window shown below should appear. (The window may be reduced initially — if so, expand it to provide some working space.)

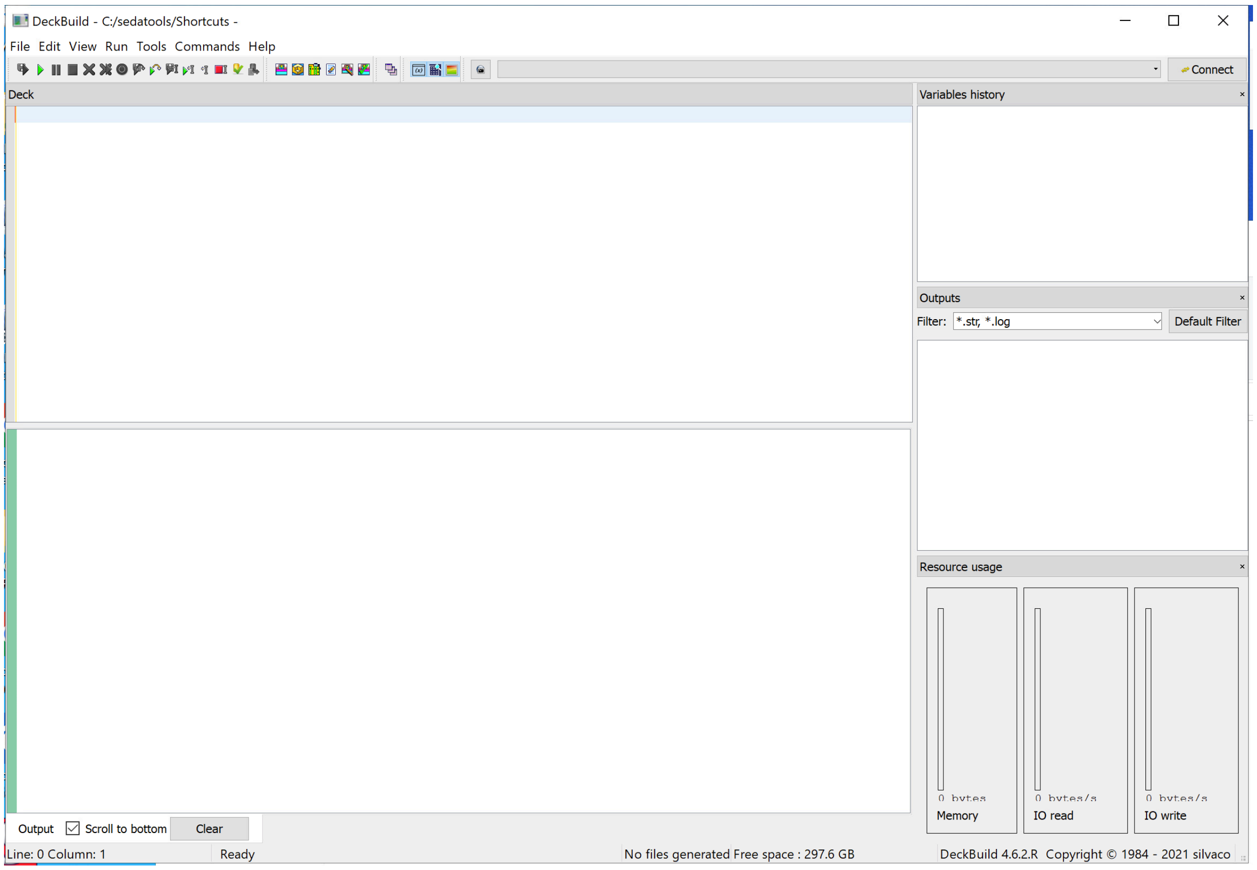


Figure 1. The Deckbuild window.

Basics of running SUPREM via Deckbuild

The Deckbuild window has two primary sections. The text with the list of commands is entered in the upper sub-window, labeled Deck. The lower sub-window, labeled Output, displays status messages and results as the simulation runs. The small areas to the right are used to show some results and other information.

The input to a SUPREM simulation is a simple text file, consisting of a series of command lines. Generally, each command line corresponds to one step in the fabrication process. We can type the text for most of the commands directly into the "Deck" area of the program window. Alternatively, we could type everything into a separate file using a text editor, save it on disk, and then open the text file in Deckbuild. For the initial tutorial exercises, it is probably easiest to type directly into the Deckbuild window.

The simple examples below will show the basic syntax of commands that will be useful for EE 432/532 work. Also, Silvaco provides some example input files — see "Examples…" under the File menu— that can be examined to see command syntax. Of course, there is always the manual — choose "Deckbuild Help" under the Help menu. We might note that in some of the examples, certain keywords can be shortened. For example, "loc" for location, "spac" for spacing, "phosphor" for phosphorus, etc. It is not necessary to use the abbreviations — it is our choice. If uncertain about abbreviations, it is probably safest to just spell everything out.

A basic input files includes four parts:

1. A definition of the grid,
2. an initiation statement to define the wafer doping type and level (and possibly other info),
3. the commands for the actual processing steps to be simulated, and
4. commands for extracting various output variables.

As with any program, explanatory comments are helpful. In the input file, comment lines are denoted with a #-symbol. These lines will be ignored during when the simulation runs.

Once the input file is complete, the simulation can be started. The simulation flow is controlled using the buttons just above the upper window. The simulation can run in a line-by-line mode or the whole file can be run in one shot. In either case, copious amounts of (sometimes mysterious) output will appear in the lower window. For the most part, we can ignore most of it. However, the results of any extraction command will appear in the output area along with all of the run-time status text, and we will need to watch for it. (Usually, it is highlighted). In addition the extracted value will also be printed in the small "Variables history" window in the upper right — with too many significant digits and no units. (Urk.) Finally, we can plot the results using an auxiliary program called TonyPlot — more on that later.

Starting and stopping

The first line in the input deck should be "`go Athena`". This launches the simulation program. (Athena is the original name of the software that was developed at Stanford University.)

The final line should be `quit`, for obvious reasons.

Grid definition

SUPREM solves the fabrication equations for diffusion, oxidation, implantation, etc. using finite-element techniques, and that requires us to define a grid of points within the region of interest. The various equations will be solved at each of those points. As is always the case with finite-element computations, there is a trade-off between grid density and computation speed. Defining a very dense array of grid points will allow us to generate smooth curves, but the computation will take longer. Defining a relatively sparse grid will speed up the computation but may lead to "ragged" plots or the possibility of missing interesting things that are happening on length scales smaller than the spacing between grid points.

SUPREM is inherently a two-dimensional solver, and so we must define a two-dimensional grid. The x-dimension is parallel to the wafer surface and the y-dimension is perpendicular — i.e. into the wafer. (Note that this is opposite to the orientation used in the equations in the class notes, where the x-direction was "into" the wafer." The grid definition commands specify the grid at particular points. SUPREM then fills in more points based on the listed points. For example, the set of commands:

```
line x loc=0.00 spacing=0.10
line x loc=5.00 spacing=0.10

line y loc=0.00 spacing=0.05
line y loc=5.00 spacing=0.05
```

defines the corners of a square space ranging from $0 \leq x \leq 5$ μm and from $0 \leq y \leq 5$ μm. The commands "suggest" grid spacings of 0.1 μm in the x-direction and 0.05 μm in the y-direction to fill the space. SUPREM will use these defined corner points to fill entire the computational space with grid points, with nominal x-spacing of 0.1 μm and y-spacing of 0.05 μm.

The grid spacing does not have to be uniform. We can define a dense grid in regions where doping profiles are changing rapidly and a sparser grid in the regions where changes are more gradual. This allows for more accuracy in the interesting regions and more speed in the regions that are less interesting. By including more grid commands, we can have SUPREM define regions of denser and sparser grids points. For example, the commands

```
line x loc=0.00 spacing=1.00
line x loc=1.00 spacing=1.00

line y loc=0.00 spacing=0.05
line y loc=2.00 spacing=0.05
line y loc=5.00 spacing=0.2
```

will lead to a denser grid spacing of 0.05 μm in the region $0 \leq y < 2$ μm, and then a sparser spacing between 2 μm and 5 μm. This works because we expect most of the variation near the surface while deeper down the concentrations are changing less rapidly. When quantities are changing less rapidly, we can need fewer points in order to get smooth curves. Note in the example that there will be only two grid points in the x-direction, $x = 0$ and $x = 1.0$ μm. This is a way of defining a "quasi-1D" problem. We can use this trick when we are interested only in what happens in the y-direction. (As is often the case for EE 432/532 problems.) A "quasi-1D" simulation will gives results similar to the one-dimensional calculations done in class using the ideal equations.

SUPREM has an adaptive method for determining grid points between the defined points, so the grid may not be exactly what we would expect based on the corner point definitions. As hinted above, SUPREM treats the spacing listed in the command line as a "suggestion". (See the manual to dig into the details.) Usually, this is not a big issue.

Substrate initialization

The next step is a single line to define the starting substrate. We can specify many things about the wafer, but the most important are the type and the doping. Obviously, all simulations for 432/532 will use silicon as the material for the substrate. (SUPREM can handle other semiconductors.) The substrate doping is assumed to be uniform, and we must specify the type and doping level. The doping level can be specified by concentration or resistivity — resistivity is converted to the equivalent concentration. We can also specify the orientation, although the default is the (100) surface, which is by far the most commonly used starting wafer surface.

Substrate command examples:

# (100) silicon doped with boron at 10^{15} cm^{-3}
```
init silicon c.boron=1.0e15 orientation=100
```

# (100) silicon doped with phosphorus and having a resistivity of 5 Ω·cm.
```
init silicon phosphorus resistivity=5 orientation=100
```

Fabrication commands

Next up are the fabrication command lines. Each command line describes a specific step. Note that all high temperature steps — including oxidations — are called “diffusions”. The command line includes any parameters that are needed specify the particular fab step. The commands should be entered in the correct order (for obvious reasons), and so can be made to look almost like a process traveler.

Examples of command lines

# Dry oxidation at 1000°C for 60 minutes
```
diffuse time=60 temp=1000 dryo2
```

# Wet oxidation at 1100°C for 30 minutes
```
diffuse time=30 temp=1100 weto2
```

# Constant-source phosphorus diffusion, 1000°C for 30 minutes in a nitrogen ambient with
#constant surface concentration = 10^{21} cm^{-3}
```
diffuse time=30 temp=1000 nitrogen c.phosphor=1.0e21
```

(The c.phosphor parameter sets the surface concentration of the constant-source diffusion. In this case, c.phosphor is chosen to be equal to the solid-solubility limit of phosphorus in silicon at 1000°C.)

# Temperature ramp from 800°C to 1100°C in 20 minutes with nitrogen ambient
```
diffuse time=20 temp=800 t.final=1100 nitro
```

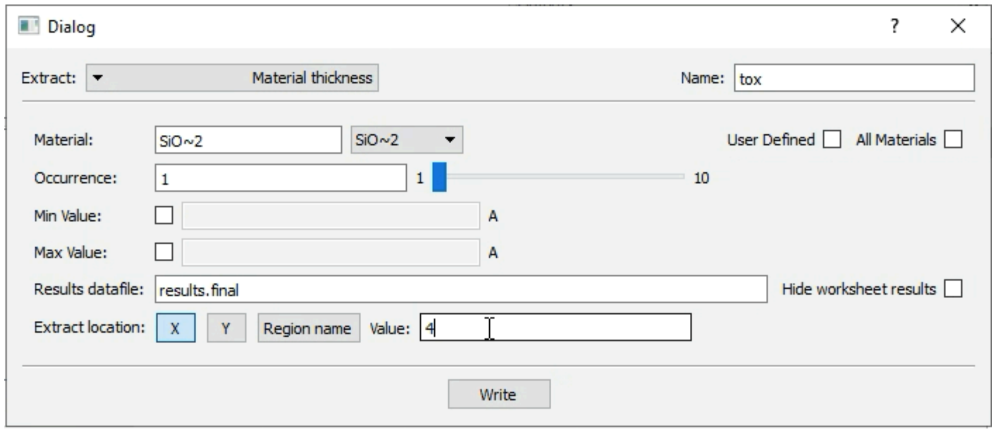
# Boron implant at 100 keV and dose of 10^{14} cm^{-3}
```
implant boron dose=1.0e14 energy=100
```

There are also commands for lithography, etching, deposition of many different types (including oxides, nitrides, and polysilicon), and growing silicon epitaxial layers.

The full capability of SUPREM becomes evident in setting up the process steps. Things that we could only approximate or discuss qualitatively (like temperature ramps, multiple oxidation/ diffusion steps, lateral diffusions, and segregation effects) are handled with ease by SUPREM. Plus, we have control over all kinds of parameters relating to the process steps. We won’t be looking at these in this short tutorial — mostly we will stick with default values. All parameters are described in the manual.

Extraction commands

Various quantities can be calculated from the simulation results. Of typical interest are things like oxide thicknesses, junction depths, sheet resistances, and MOS threshold voltages. These are defined by extraction commands and the results are printed in the lower "Output" sub-window, mixed in with all the text as the simulation runs. Also, the results are displayed in the "Variables history" sub-window. Extraction command syntax is a bit tricky. Fortunately extraction commands can be generated using a dialog. Use the "Commands —> Athena Extract" menu item to bring up the dialog box. Using the dialog helps ensure that the syntax is correct. The figure below shows on example of the extract dialog, filled out to find oxide thickness.



Extract: Material thickness
Name: tox
Material: SiO~2 SiO~2
User Defined ☐ All Materials ☐
Occurrence: 1 1 10
Min Value: ☐ A
Max Value: ☐ A
Results datafile: results.final
Hide worksheet results ☐
Extract location: X Y Region name Value: 4

The extraction command always needs a name that is unique to the particular extraction. Also, we must specify an x-value within the grid where the particular value will be extracted. (For example, since oxide thickness is measured by taking a slice in the y-direction, we must tell the simulation at what value of x to take the vertical slice.) Also, since there can be multiple layers or junctions, the command may also need an "occurrence number". The item closest to the surface is 1 and the occurrence increments going deeper into the wafer. Clicking the "Write" button in the dialog causes the extraction command to be written into the file:

```
extract name=“tox” thickness material=“SiO~2” mat.occno=1 X.val=4
```

Other extraction-command examples:

```
# Find the depth of the first junction.
extract name="xj1" xj material="Silicon" x.val=0.5 junc.occno=1
```

Note:"xj" is the command for finding junction depth, although the depth is in the y-direction. For the second junction, we would use a different name and change "junc.occno" to 2.

```
#Find the sheet resistance of the top-most doped layer.
extract name="Rs" sheet.res material="Silicon" x.val=0.5 \ region.occno=1
```

```
#Find the surface concentration of a doped layer
extract name="N(0)" surf.conc impurity="Net Doping" material="Silicon"
mat.occno=1 x.val=0.5
```

Note that if a line is too long to fit in the text window. It will wrap automatically.

Also, some device parameters can be extracted.

```
#Calculate the threshold voltage of an p-type MOS structure.
extract name="PMOS VT" 1dvt ptype x.val=1.0

#Calculate the forward beta for an npn BJT structure.
extract name="npn bf" bf x.val=1.0
```

Saving and re-loading files

The input file must be saved before the simulation will run. Then, if needed, we can re-run the simulation again at a later time. Or we can use a previously saved file to serve as a starting template for making a new simulation file.

The expected suffix for input files ".in".

The default location for saving files is our home directory (C:\Users\gtuttle for the current example). However, files can be stored in other locations — we can make a sub-folder or specify another directory, as we see fit.

To load a previously saved file, use "File —> Open…" and navigate to the location of the stored file.

When quitting Deckbuild or when switching to another file, we may receive a warning about a hidden folder with history files and a question above removing them. These are temporary files that the software writes as the simulations are being run. These history file are what allow us to stop and start simulations or jump back to a different point in the execution sequence. Unless we are running a particularly large and time-consuming simulation, it is probably OK to delete those files. For the simple simulations that typical in 432/532, the history file information can be generated again easily.

The files needed for making plots are also in the same directory location as the input file. (See below.)

Plotting

SUPREM works with an associated plotting program called *TonyPlot*. In principle, we could insert a `tonyplot` command anywhere within a command list and a plot of the structure and doping profiles would be pop up automatically when the execution hits that point. We could generate one plot at the end or twenty different plots along the way — whatever helps us best understand the fabrication process.

However, the virtual machine implementation of the Silvaco tools seems to interfere with the automatic generation of plots. As each command is executed, a "structure history file" containing all the current process information is stored somewhere on the disk. Then, when a `tonyplot` command comes up, the TonyPlot program should use the current history file to generate the plot.

However, with the virtual machine set up, it seems that TonyPlot cannot find the necessary files. When the simulation reaches the `tonyplot` command, the plotting software will be launched, but nothing will happen.

The work-around for this problem is to force a *structure file* to be written at the point (or points) where we would like to make a plot. The syntax is

```
structure outfile=plot_file.str
```

where "plot_file.str" file will be written in the same location as the input file is saved. An icon linked to the structure file will appear in the output section of the simulation. Then, once the simulation has finished, we can double-click on that icon to launch TonyPlot and see the plot.

If we want to make several plots at different points in the process, we must give each structure file a unique name, otherwise the previously generated file will be overwritten.

There are several TonyPlot graphs shown in the examples below.

Examples

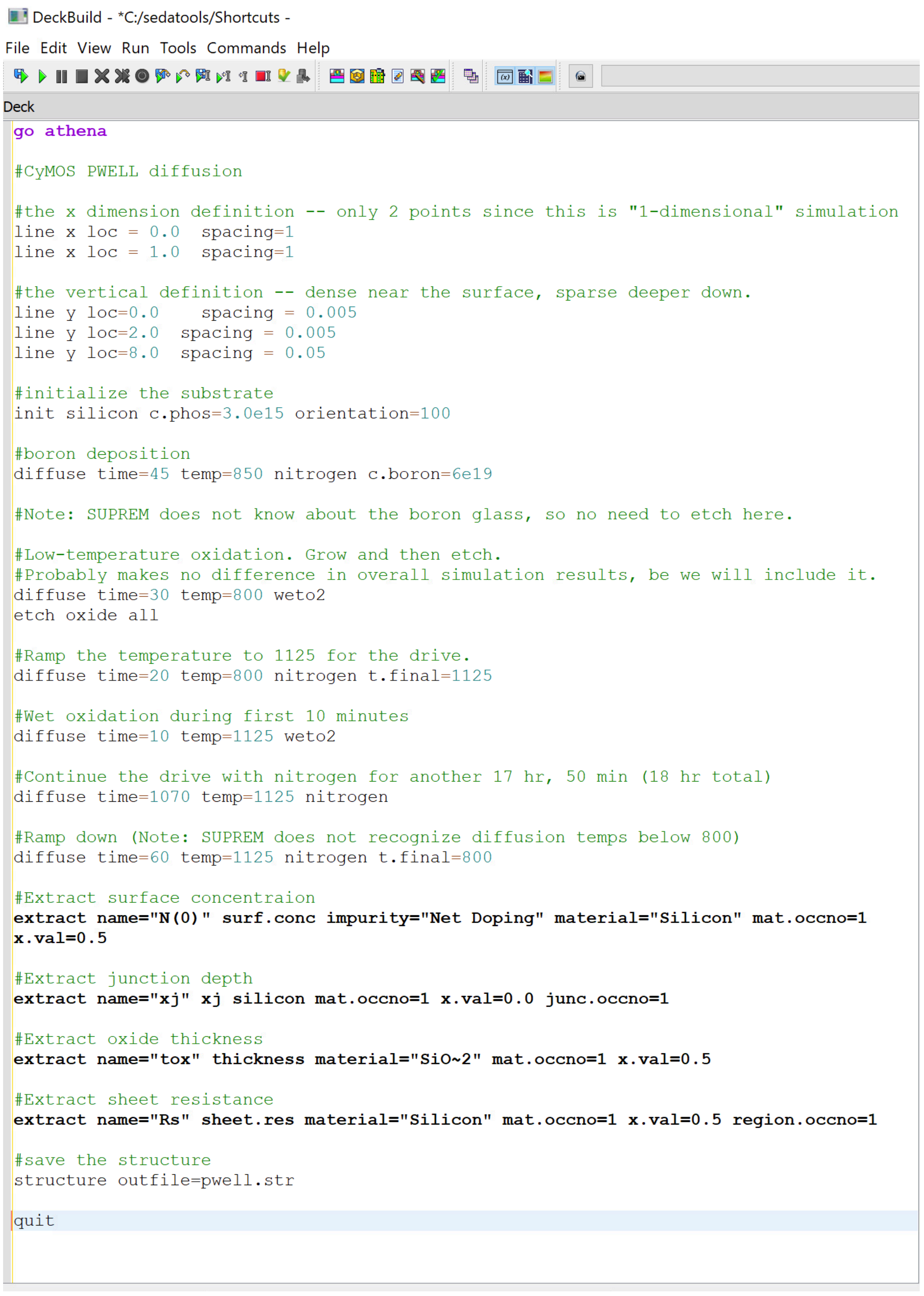
Below are a few examples of EE 432/532 type simulations. Looking through these and maybe running one or two is a good way to become familiar with SUPREM,

Example 1 - PWELL diffusion

Below is a “quasi-1D” simulation file for p-well boron diffusion used in the CyMOS process.

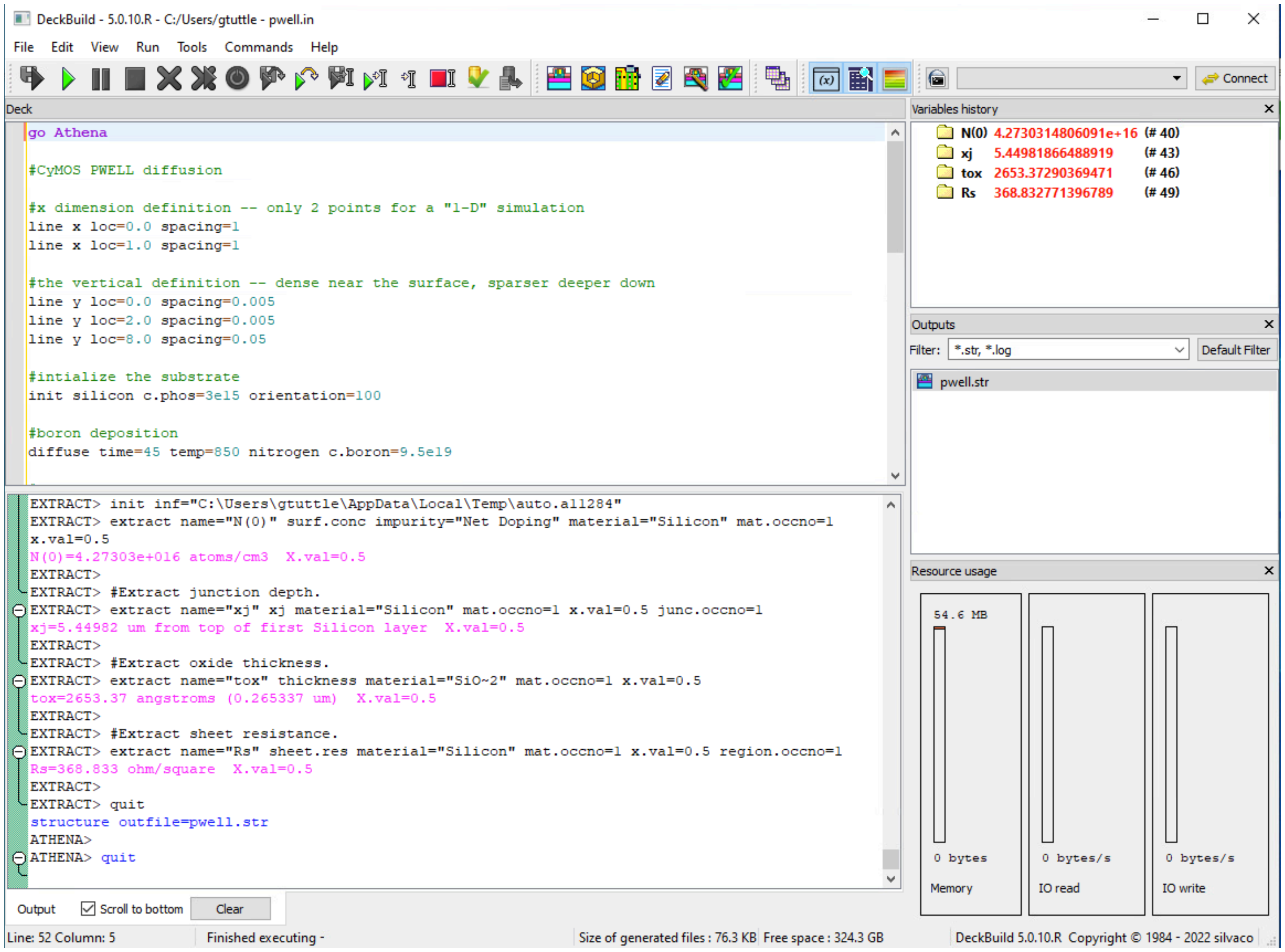
```
#CyMOS PWELL diffusion

go athena

#x dimension definition -- only 2 points for a “1-D” simulation
line x loc=0.0  spacing=1
line x loc=1.0  spacing=1

#the vertical definition -- dense near the surface, sparse deeper down.
line y loc=0.0  spacing = 0.005
line y loc=2.0  spacing = 0.005
line y loc=8.0  spacing = 0.05

#initialize the substrate
init silicon c.phos=3.0e15 orientation=100

#boron deposition
diffuse time=45 temp=850 nitrogen c.boron=9.5e19

#Note: SUPREM does not know about the boron glass, so no need to etch here.

#Low-temperature oxidation. Grow and then etch.
#Probably makes no difference in overall simulation results, but include anyway.
diffuse time=30 temp=800 weto2
etch oxide all

#Ramp the temperature to 1125 for the drive.
diffuse time=20 temp=800 nitrogen t.final=1125

#Wet oxidation during first 10 minutes
diffuse time=10 temp=1125 weto2

#Continue the drive with nitrogen for another 17 hr, 50 min (18 hr total)
diffuse time=1070 temp=1125 nitrogen

#Ramp down (Note: SUPREM does not recognize diffusion temps below 800)
diffuse time=60 temp=1125 nitrogen t.final=800

#Extract surface concentration
extract name="N(0)" surf.conc impurity="Net Doping" material="Silicon" \
mat.occno=1 x.val=0.5

#Extract junction depth
extract name=“xj" xj material=“Silicon” mat.occno=1 x.val=0.0 junc.occno=1

#Extract oxide thickness
extract name="tox" thickness material="SiO~2" mat.occno=1 x.val=0.5

#Extract sheet resistance
extract name="Rs" sheet.res material="Silicon" mat.occno=1 x.val=0.5
region.occno=1

#save the structure for plotting
structure outfile=pwell.str

quit
```

Figure 2. PWELL simulation commands in Deckbuild. (The Output sub-window has been reduced to make more room to see the Deck. Sub-windows can be resized by clicking on and dragging the dividing borders.)


DeckBuild - *C:/sedatools/Shortcuts -
File Edit View Run Tools Commands Help
Deck

```
go athena

#CyMOS PWELL diffusion

#the x dimension definition -- only 2 points since this is "1-dimensional" simulation
line x loc = 0.0  spacing=1
line x loc = 1.0  spacing=1

#the vertical definition -- dense near the surface, sparse deeper down.
line y loc=0.0    spacing = 0.005
line y loc=2.0  spacing = 0.005
line y loc=8.0  spacing = 0.05

#initialize the substrate
init silicon c.phos=3.0e15 orientation=100

#boron deposition
diffuse time=45 temp=850 nitrogen c.boron=6e19

#Note: SUPREM does not know about the boron glass, so no need to etch here.

#Low-temperature oxidation. Grow and then etch.
#Probably makes no difference in overall simulation results, be we will include it.
diffuse time=30 temp=800 weto2
etch oxide all

#Ramp the temperature to 1125 for the drive.
diffuse time=20 temp=800 nitrogen t.final=1125

#Wet oxidation during first 10 minutes
diffuse time=10 temp=1125 weto2

#Continue the drive with nitrogen for another 17 hr, 50 min (18 hr total)
diffuse time=1070 temp=1125 nitrogen

#Ramp down (Note: SUPREM does not recognize diffusion temps below 800)
diffuse time=60 temp=1125 nitrogen t.final=800

#Extract surface concentraion
extract name="N(0)" surf.conc impurity="Net Doping" material="Silicon" mat.occno=1
x.val=0.5

#Extract junction depth
extract name="xj" xj silicon mat.occno=1 x.val=0.0 junc.occno=1

#Extract oxide thickness
extract name="tox" thickness material="SiO~2" mat.occno=1 x.val=0.5

#Extract sheet resistance
extract name="Rs" sheet.res material="Silicon" mat.occno=1 x.val=0.5 region.occno=1

#save the structure
structure outfile=pwell.str

quit
```

Figure 3. PWELL simulation with some of the output text.



```
go Athena

#CyMOS PWELL diffusion

#x dimension definition -- only 2 points for a "1-D" simulation
line x loc=0.0 spacing=1
line x loc=1.0 spacing=1

#the vertical definition -- dense near the surface, sparser deeper down
line y loc=0.0 spacing=0.005
line y loc=2.0 spacing=0.005
line y loc=8.0 spacing=0.05

#intialize the substrate
init silicon c.phos=3e15 orientation=100

#boron deposition
diffuse time=45 temp=850 nitrogen c.boron=9.5e19
```

```
N(0)  4.2730314806091e+16  (# 40)
xj    5.44981866488919     (# 43)
tox   2653.37290369471     (# 46)
Rs    368.832771396789     (# 49)
```

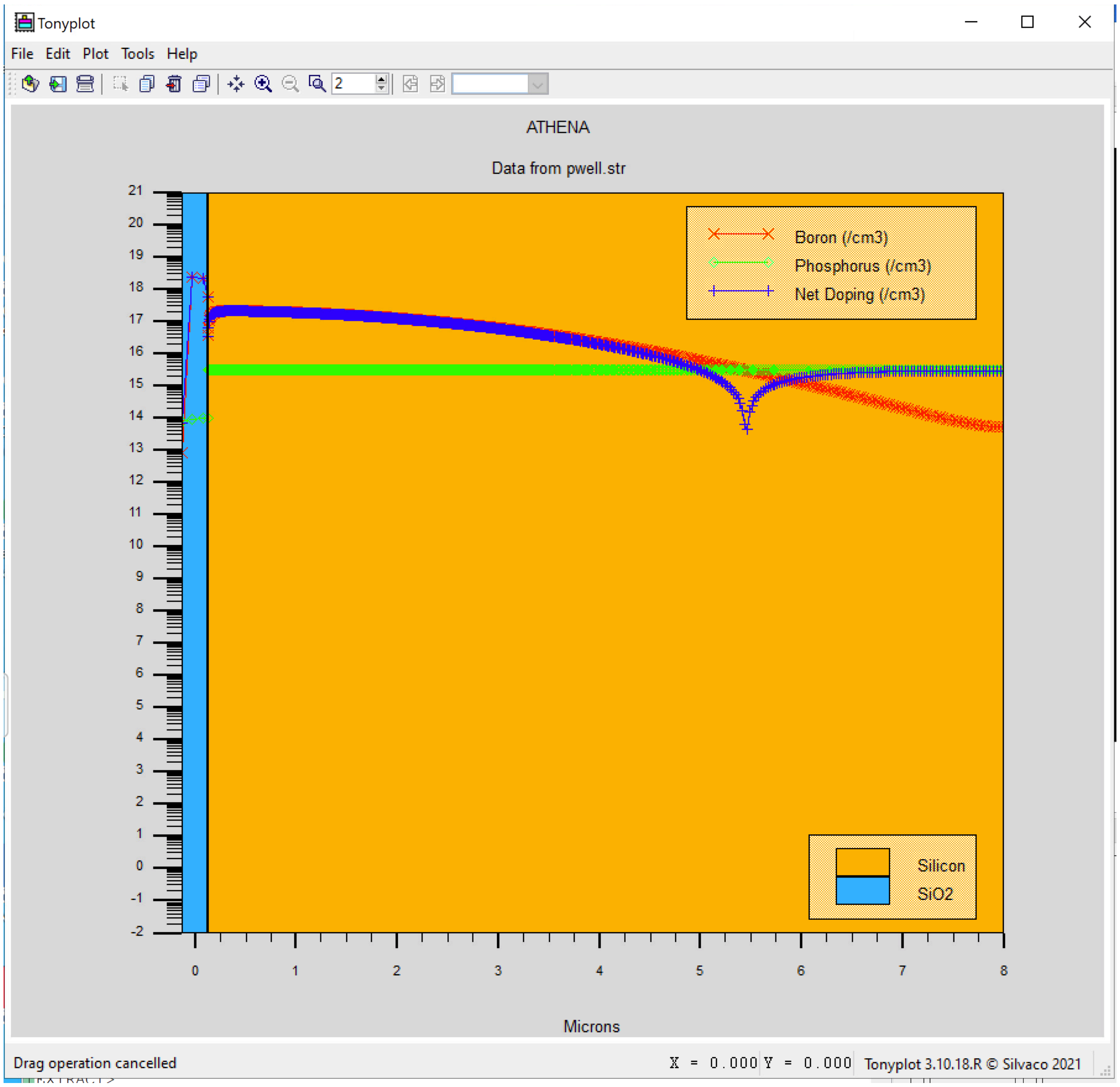
```
EXTRACT> init inf="C:\Users\gtuttle\AppData\Local\Temp\auto.a11284"
EXTRACT> extract name="N(0)" surf.conc impurity="Net Doping" material="Silicon" mat.occno=1
x.val=0.5
N(0)=4.27303e+016 atoms/cm3  X.val=0.5
EXTRACT>
EXTRACT> #Extract junction depth.
EXTRACT> extract name="xj" xj material="Silicon" mat.occno=1 x.val=0.5 junc.occno=1
xj=5.44982 um from top of first Silicon layer  X.val=0.5
EXTRACT>
EXTRACT> #Extract oxide thickness.
EXTRACT> extract name="tox" thickness material="SiO~2" mat.occno=1 x.val=0.5
tox=2653.37 angstroms (0.265337 um)  X.val=0.5
EXTRACT>
EXTRACT> #Extract sheet resistance.
EXTRACT> extract name="Rs" sheet.res material="Silicon" mat.occno=1 x.val=0.5 region.occno=1
Rs=368.833 ohm/square  X.val=0.5
EXTRACT>
EXTRACT> quit
structure outfile=pwell.str
ATHENA>
ATHENA> quit
```

The results of the extraction commands can be plucked out of the info displayed in the Output window during the run. The same information is displayed in the area at upper right. (But with no units.)

```
N(0) = 4.27303e+16 atoms/cm3 X.val=0.5

xj=5.44982 um from top of first Silicon layer  X.val=0.5

tox=2653.37 angstroms (0.26557 um)  X.val=0.5

Rs=368.833 ohm/square  X.val=0.5
```

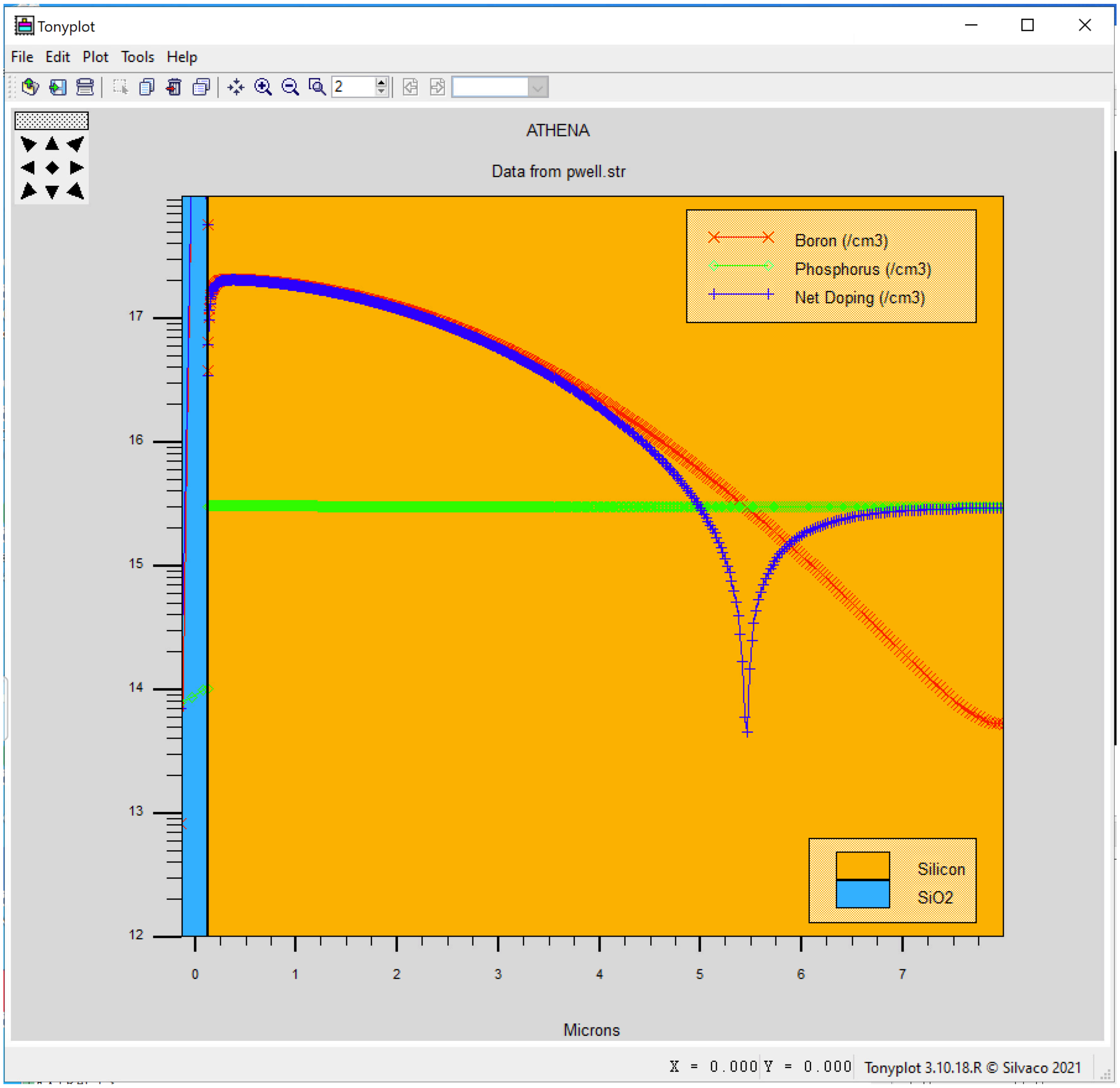
To make a plot of the oxide layer and doping profile in the silicon, double-click on the "pwell.str" icon in the "Outputs" section on the right side of the window. (There will be momentary pause while the TonyPlot software is launched. Be patient and don't double-click again too quickly or there will a second version of the plot once the software starts up!) Again, the structure is stored in the same directory as the SUPREM input text file.

Figure 4. Graph of the PWELL simulation results, prior to re-sizing.



There are many controls for manipulating the graph, adding text and line art, and performing other calculations. (Use the "Help" menu to bring up the TonyPlot manual to explore further.) The one thing we should probably do for every graph is to resize the axes to improve readability. In the graph above, there are about 15 orders of magnitude included in the vertical scale that really aren't needed. Choose the "Set Zoom…" item under the "Plot" menu. A dialog appears where the starting and ending x and y coordinates can be set. Change "Starting Coord Y" to 12 or 13 and change "Ending Coord Y" to 18 or 19. (Note that these number are exponents for the vertical log axis.) Then click "OK" to re-plot the graph with more reasonable limits. We could change the x-axis limits as well, but there is really need in this case.

Figure 5. PWELL graph after re-sizing.



Example 2 - Simple NMOS threshold voltage calculation

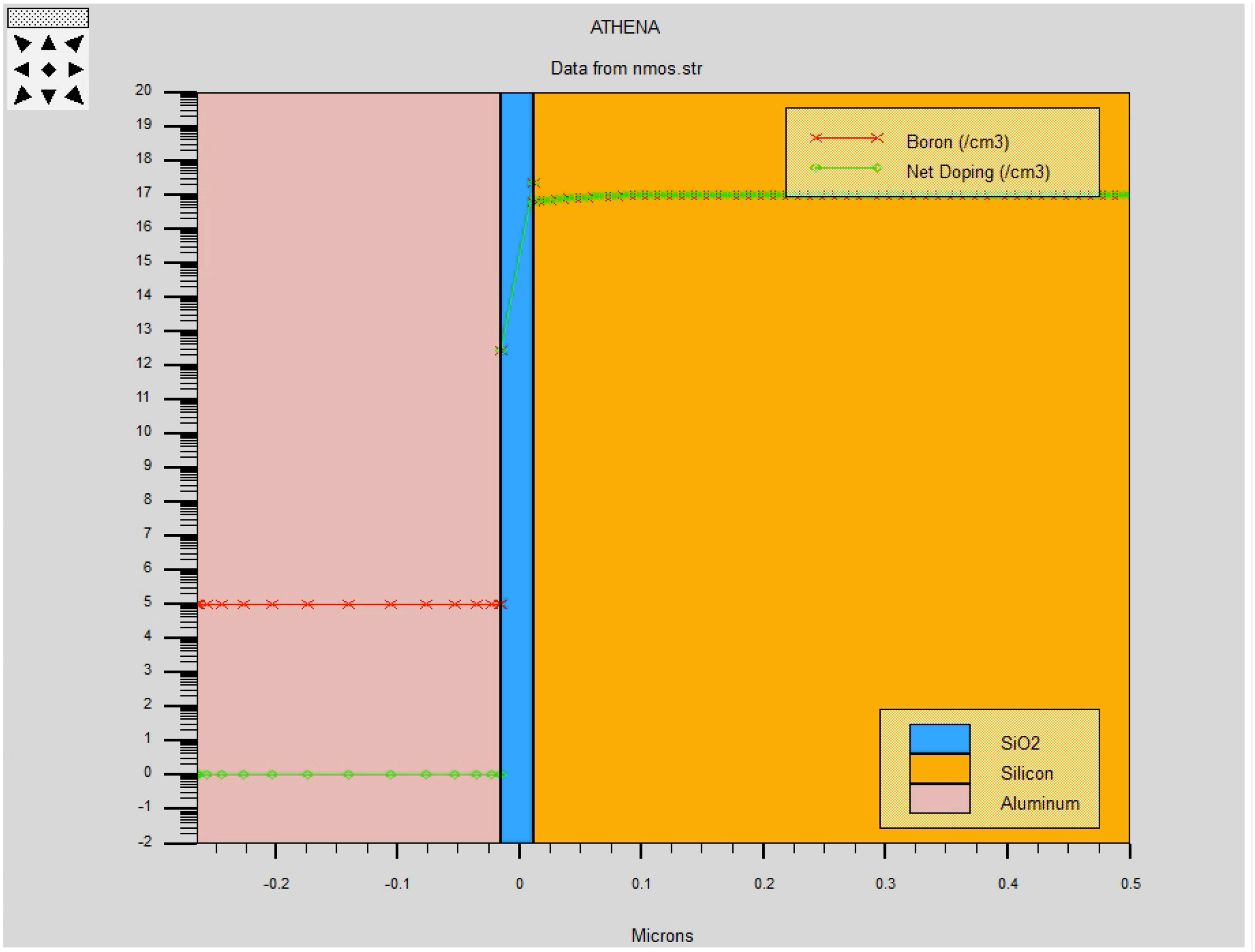
Here is the listing for a simulation to determine the expected threshold voltage for an NMOS transistor. The extraction statement uses the “QICKMOS 1D Vt” option from the “Commands—>Athena Extract…” menu item. The threshold calculation uses a simplified model for the threshold voltage, so it may not accurately predict real device parameters, but it can give us an idea of how V_T may vary with fabrication conditions.

```
#NMOS threshold voltage example

go athena

#define a simple 1x3 grid, sparse in x and denser in y
line x loc=0.0  spacing=0.1
line x loc=1.0  spacing=0.1

line y loc=0.0  spacing=0.005
line y loc=2.0  spacing=0.005

#initialize the substrate, boron at 1e17
init silicon c.boron=1.0e17 orientation=100

#Grow an oxide under dry conditions. Aim for 25 nm
diffuse time=20 temp=1000 dryo2

#Extract oxide thickness
extract name="tox" thickness material="SiO~2" mat.occno=1 x.val=0.5

#Put on a metal gate, 0.25 microns thick
deposit aluminum thickness=0.25

#Extract NMOS threshold voltage
extract name=“NMOS VT" 1dvt ntype x.val=0.5

#save the structure for plotting
structure outfile=nmos.str

quit
```

Results from the displayed output:

```
   tox=258.96 angstroms (0.025896 um)  X.val=0.5

   NMOS VT=0.953327 V  X.val=0.5
```

Figure 6. The plot of the NMOS structure. The x-axis max (Ending Coord X) was changed to 0.5 microns. The plot is not very interesting — there isn't much going on.



Example 3 - Boron ion implantation

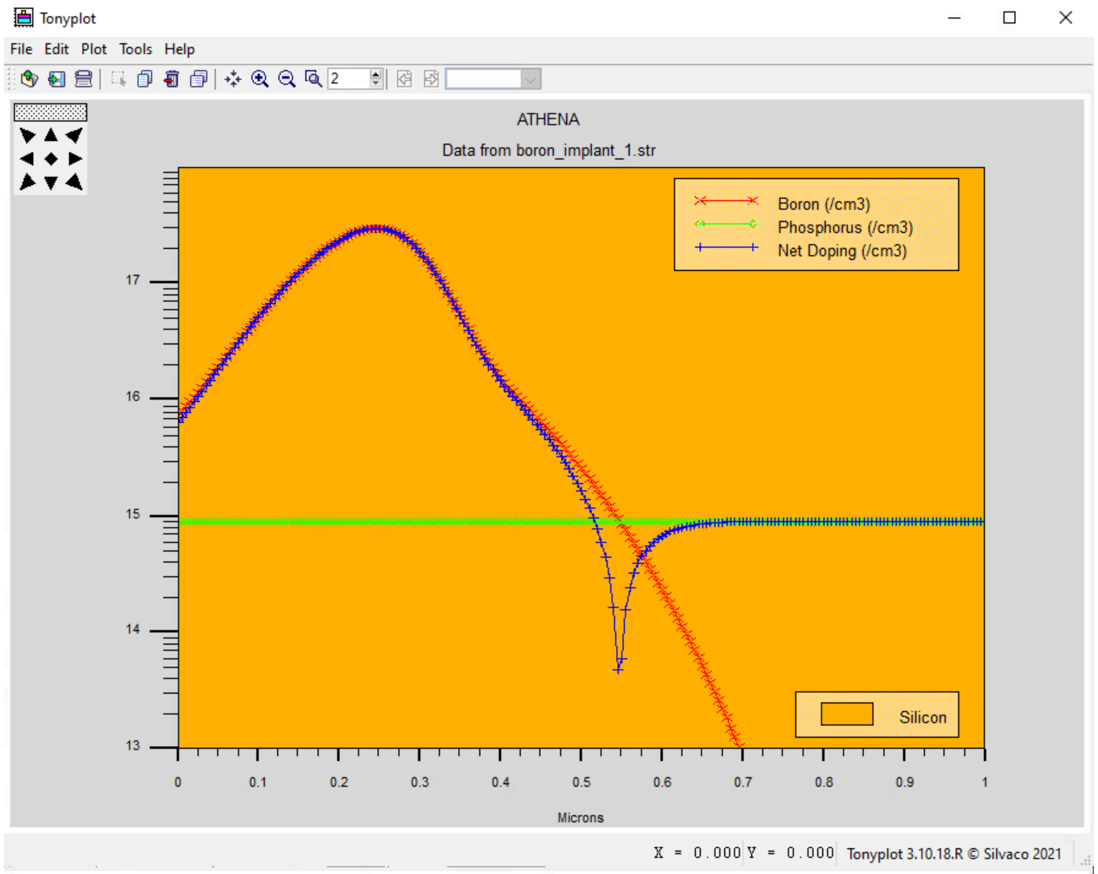
Here is a listing for the simulation of a boron implantation into an n-type silicon wafer. The implant parameters are E = 75 keV and $Q = 5 \times 10^{12}$ cm^{-2}. We extract the junction depth immediately after the implant. The wafer is then annealed (T = 1000°C for 30 minutes in a wet oxidizing environment), as is necessary after an implant, and the junction depth is extracted along with the thickness of the oxide that was grown during the anneal. Structure files are saved at both points so that we can compare the profiles after the implant and after the anneal.

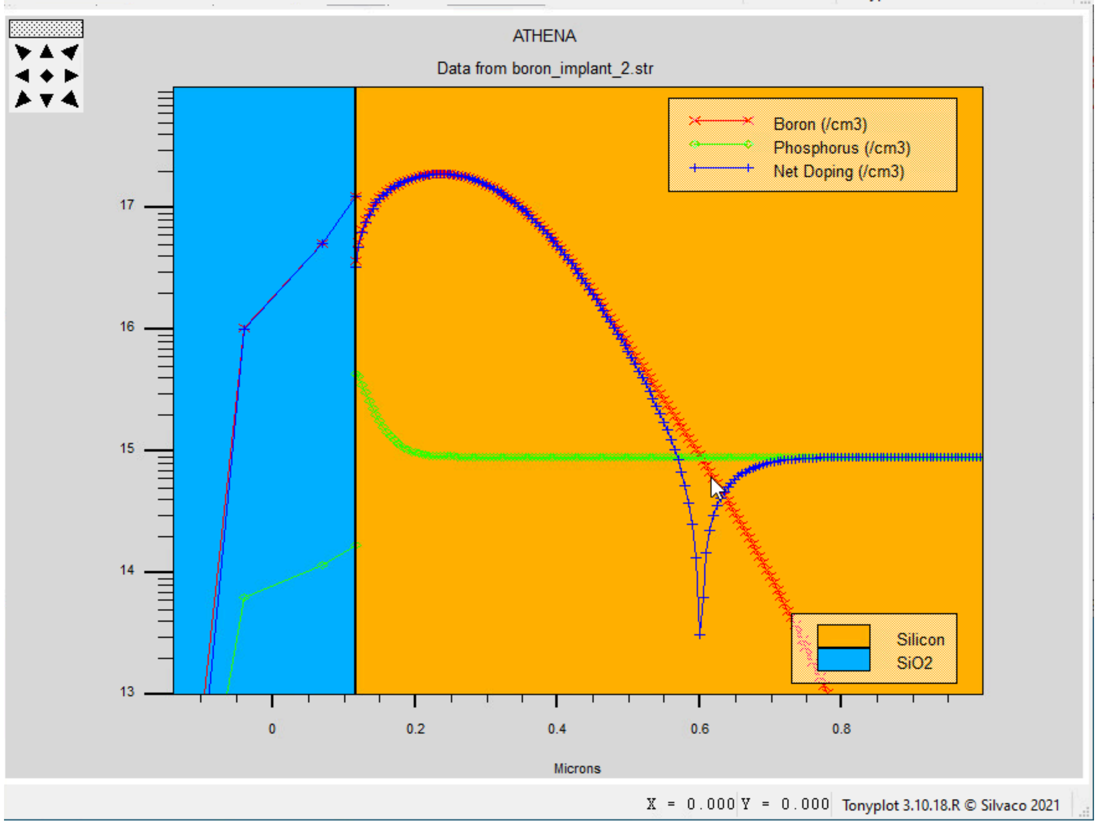
```
#Boron implant example

go athena

#define a simple "1-D" grid with varying density in y
line x loc=0.0  spacing=1.0
line x loc=1.0  spacing=1.0

line y loc=0.0  spacing = 0.005
line y loc=1.0  spacing = 0.005
line y loc=3.0  spacing = 0.05

#initialize the substrate, phosphorus doped with resistivity = 5 ohm-cm
init silicon phosphorus resistivity=5 orientation=100

#implant boron at 75 keV and dose = 5e12 cm^-2
implant boron energy=75 dose=5e12

#Extract junction depth now
extract name="xj pre anneal" xj material="Silicon" mat.occno=1 x.val=0.0 \
junc.occno=1

#save the as-implanted
structure outfile=boron_implant_1.str

#Anneal the implanted wafer while growing a wet oxide
diffuse time=30 temp=1000 weto2

#Extract junction depth again, post anneal
extract name="xj post anneal" xj material="Silicon" mat.occno=1 x.val=0.0 \
junc.occno=1

#Extract oxide thickness
extract name="tox" thickness material="SiO~2" mat.occno=1 x.val=0.5

#save the annealed profile
structure outfile=boron_implant_2.str

quit
```

Results from the displayed output:

```
  xj pre anneal=0.547153 um from the top of the first silicon layer  X.val=0.5

  xj post anneal=0.485341 um from the top of the first silicon layer  X.val=0.5

  tox=2577.64 angstroms (0.257764 um)  X.val=0.5
```

Figure 7. Boron profile after implant (top) and after anneal (bottom)


Tonyplot
File Edit Plot Tools Help
ATHENA
Data from boron_implant_1.str
Boron (/cm3)
Phosphorus (/cm3)
Net Doping (/cm3)
Silicon
17
16
15
14
13
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Microns
X = 0.000 Y = 0.000
Tonyplot 3.10.18.R © Silvaco 2021


ATHENA
Data from boron_implant_2.str
Boron (/cm3)
Phosphorus (/cm3)
Net Doping (/cm3)
Silicon
SiO2
17
16
15
14
13
0
0.2
0.4
0.6
0.8
Microns
X = 0.000 Y = 0.000
Tonyplot 3.10.18.R © Silvaco 2021