

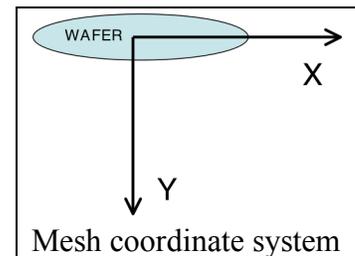
Using TSUPREM-4 for Piezoresistor Design

M. Hopcroft MAY2006 v1.01

TSUPREM-4 can be used for predicting the characteristics of piezoresistors created by doping of silicon. However, some care must be taken with the design of the input file and the interpretation of the output in order to get meaningful results. This document describes the current best practices for a one-dimensional simulation that predicts the characteristics of a rectangular piezoresistor. The relevant parameters for piezoresistor design that we wish to extract from TSUPREM-4 are: sheet resistance, R_s , junction depth, D_j , peak carrier concentration, N_p , and total number of carriers, N . In this document, numbers in square brackets, [X-XXX], refer to page numbers in the TSUPREM-4 manual, version 2002.4, February 2003. TSUPREM-4 commands are shown in fixed-width font, e.g., `initialize`.

Define Mesh

TSUPREM-4 begins a simulation by defining a mesh of elements and node points for the calculations (aka “bottom-up modeling”). All TSUPREM-4 simulation files are “two-dimensional”, but the calculations are performed in one dimension as long as the structure has 1-D uniformity [2-12]. Our simulation will consider the Y-direction only (into the wafer). We can use the default mesh in the x-direction, which is the minimum 2 lines wide ([0 1]). The default y-mesh, however, is too coarse, so we specify it to have 0.01 μm spacing near the surface, where the action is.



```
line y      loc=0.0  spacing=0.01    tag=top
line y      loc=10.0 spacing=0.10    tag=bottom
```

Type of Boundaries

There are two possible boundary conditions: *reflecting* and *exposed*. Exposed is appropriate for the surface of the wafer where oxidation and ion implants occur. Reflecting is appropriate for internal boundaries. The default condition for boundaries other than the top is *reflecting*, so no Boundary statement is required for these simulations [3-58].

Type of Region

The default region type is silicon, so no region statement is required for these simulations [3-60].

Initialize Material

The `initialize` statement determines the initial background doping level of the wafer. The doping can be specified in terms of a resistivity or doping level.

```
initialize <100> impurity=phosphorus i.resistivity=5.00
```

Calculation Method

The method statement specifies the manner of the calculations that are performed. There are a large number of options [3-183]. The important parameters for piezo simulations are `vertical`, specifying a 1-D oxidation model, and `pd.full`, which is the most accurate (and time-consuming) model of dopant diffusion.

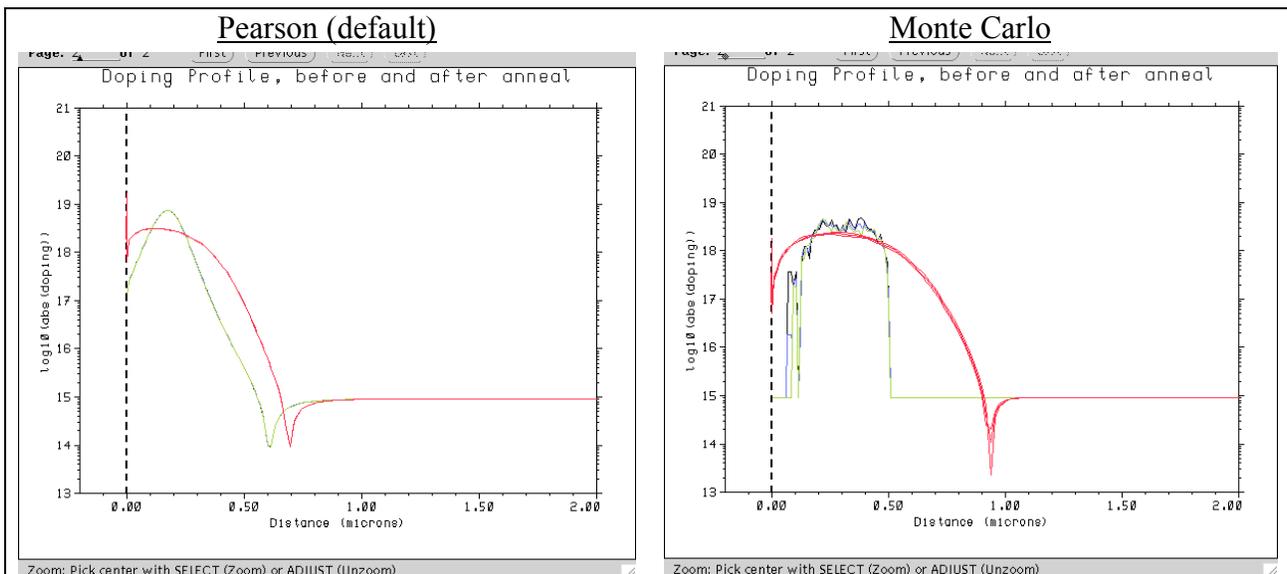
```
method      vertical pd.full
```

Ion Implant

The `implant` statement describes the ion implant by dose, energy, and angle. The modeling of ion implants is extremely complex and detailed [2-83]. The most important choice is between an analytic model (a Pearson distribution) of the resulting ion profile or a Monte Carlo simulation. The Monte Carlo simulation "... is physically based, and allows more general implant conditions and characteristics to be modeled" [3-106]. It is not clear how much impact the choice of implant model has on the resulting piezo characteristics. The `montecar periodic` statement specifies a vertical Monte Carlo simulation [3-107]. The default analytic (Pearson) model is used if the Monte Carlo is not specified.

```
implant      boron dose=1E14 energy=50 tilt=7 montecar periodic
```

Note that use of the Monte Carlo option appears to destroy the 1-D uniformity of the solution, so that the calculations will be performed for two dimensions and the results will be different along different vertical cross-sections of the model. In this case, the results from different cross-sections should be averaged. Compare the results below for Pearson and Monte Carlo models*.



* Note how the Monte Carlo solution produces 3 slightly different results, while the Pearson solutions are identical. Further comparisons of results from the two implant models are given in Appendix B.

Anneal

The diffusion statement causes annealing to happen. If the anneal occurs in an oxidizing ambient, then silicon oxidation will occur on the exposed surface [3-114]. It is common to specify multiple anneal steps in sequence in order to accurately model a specific furnace process. Temperature ramps can be specified in terms of ramps or final values.

```
diffusion    time= 10  temperature= 850           dryO2
diffusion    time= 30  temperature= 850 t.final=750  inert
```

SNF Tylan furnace recipes are available on the SNF website. However, these are not updated, so double-check with the Tycom disk for critical processes. The Thermco recipes are not available online.

Results

The results of the simulation can be determined in several ways. The first step is to use the select statement to specify the quantity of interest. Many quantities can be selected, including [3-123]:

```
boron       boron concentration (atoms/cm3)
phosphorus  phosphorus concentration (atoms/cm3)
doping      net active concentration (atoms/cm3)
```

These quantities can also be modified by mathematical operators, including *abs*, *log10*, and *slog10* [3-124]. TSUPREM-4 assigns negative values to p-type dopants (acceptors “reduce charge”) and positive values to n-type (donors “add charge”). The *abs* (absolute value) operator is often used to determine a doping level (carrier concentration) regardless of type and to compare doping levels on the same plot.

```
select  z=log10(abs(doping))
select  z=boron
```

For piezoresistors, we are interested in four primary results: junction depth, D_j , peak carrier concentration, N_p , total number of carriers, N , and sheet resistance, R_s . We use the statements *extract* [3-157], *print.1d* [3-127], and *electrical* [3-171].

Junction Depth

The following statements extract the junction depth, which is the depth where the background doping of the wafer and the implanted dopants cancel each other out, i.e., where the net doping is zero. We select the quantity *doping*, which is the combination of the background doping level and the implanted dopants, which have opposite sign. The location where they combine to zero in the silicon is the junction depth.

```
select z=doping
extract silicon x.val=0 value=0 d.extrac assign name=Dj
```

Peak Carrier Concentration

The following statements extract the peak carrier concentration, which is the maximum concentration of dopants in the implanted region. We select the quantity `abs(doping)`, because we want to find a maximum value, regardless of dopant type/sign. We assume that the maximum level of any dopants will happen in the implanted region.

```
select z=abs(doping)
extract silicon x.val=0 maximum val.extr assign name=Np0
```

Total Number of Carriers

The total number of carriers is the volume integral of the carrier concentration from the surface of the wafer to the junction depth and over the implanted area. We can get the integral over the vertical profile (depth) using the `extract` statement and specifying an integral from $y=0$ (in silicon only) to $y=D_j$ (junction depth).

```
select z=abs(doping)
extract silicon p1.y=0 p2.y=@Dj integral assign name=Ncm2
```

The resulting quantity, `@Ncm2`, is the line integral of the carrier concentration, and has units of N/cm^2 . It is up to the user to determine the total carriers in the area of a specific piezoresistor by multiplying by the area. TSUPREM-4 can do this for us if we include the piezoresistor area in the input file (units in cm):

```
assign name=length n.val=0.01
assign name=pwidth n.val=0.005
assign name=N n.val=@Ncm2*@plength*@pwidth print
```

The result, `N`, will be the total number of carriers in a specific volume. See [3-23] for details of using variables in TSUPREM-4.

The `print.1d` statement can also be used to show the junction depth and the line integral of carrier concentration, but it is not so convenient for use with `assign` for further calculation.

```
select      z=doping
print.1d    x.val=0 layers
```

Sheet Resistance

The sheet resistance is calculated using the `electric` statement [3-171]. This statement does not use a `select` command, and the default form of the statement will print the material layers, junction depth, and sheet resistance.

```
electric x.val=0
```

If we want to assign the sheet resistance to a variable, we have to use a more specific form of the command and specify which material region (not layer!) that we want.

```
electric resistance ext.reg=2 name=Rs
```

Plotting

The `plot.1d` statement can be used to plot quantities. Use the `^axes` `^clear` options to plot multiple lines on the same figure. Note that the title on the plot is part of the `select` statement.

```
select      z=log10(abs(doping)) title="Doping before & after anneal"  
plot.1d     x.val=0 y.min=13 y.max=21 x.min=-0.2 x.max=2.0  
diffusion   time= 10  temperature= 850                dry02  
select      z=log10(abs(doping))  
plot.1d     x.val=0  color=2 ^axes ^clear
```

Use `label` to write things on the plot, for example the sheet resistance:

```
label x=4.0 y=18.0 label="Final Sheet Resistance: @Rs ohms"
```

Appendix A

Example TSUPREM-4 input file for piezoresistor design, piezo_ex1.supr:

```
# piezo_ex1.supr
# example TSUPREM-4 input file for silicon piezoresistor design
# M Hopcroft MAY2006 hopcroft@stanford.edu

##
# Setup

# define the rectangular area of the piezoresistor for N calculations
# **units are cm** e.g., for 100 um, use n.val=0.01
# if you have a more complex geometry, break it into rectangles and run
# multiple simulations
assign name=length n.val=0.01
assign name=pwidth n.val=0.005

# Establish the mesh
# Use default X spacing for 1-D
# Specify a finer mesh in the Y-direction for more accuracy
line y      loc=0.0 spacing=0.01      tag=top
line y      loc=10.0 spacing=0.10     tag=bottom
# initialize the silicon
initialize <100> impurity=phosphorus i.resistivity=5.00

# make plots on the screen (instead of postscript file)
option device=X
# use detailed oxidation model
method      vertical pd.full

##
# Processing

# perform an implant
#implant    boron dose=1E14 energy=50  tilt=7  montecar periodic
implant     boron dose=1E14 energy=50  tilt=7

select      z=log(boron) title="Implanted Boron, before anneal"
plot.1d     x.val=0

# plot the results of the implant
select      z=log10(abs(doping)) title="Doping Profile, before and after anneal"
plot.1d     x.val=0 y.min=13 y.max=21 x.min=-0.2 x.max=2.0

# do an anneal
diffusion   time= 10  temperature= 850          dry02
diffusion   time= 30  temperature= 850 t.final=750  inert

# plot results after the anneal
select      z=log10(abs(doping))
plot.1d     x.val=0 color=2 ^axes ^clear

# print the results
select      z=doping
print.1d    x.val=0 layers
```

```

#####
## Results
#
# Determine the resulting sheet resistance
# "find the sheet resistance of the second silicon layer up from the
# bottom with 0 Volts applied"
electric resistance ext.reg=2 name=Rs

# From manual p 3-123:
# doping net active concentration (atoms/cm3)
# doping follows a sign convention: positive is n-type, negative p-type
# so use abs(doping) for plots and maximum values
select z=abs(doping)

# Determine the peak doping level
# "find the maximum [absolute] value of doping in the silicon along
# the line x=0"
extract silicon x.val=0 maximum val.extr assign name=Np0

# now select doping for comparison of n & p regions
select z=doping

# Determine the junction depth
# "find the distance from the silicon surface to the point
# where doping is equal to 0 along the line x=0", i.e., where
# the doping changes from p to n
extract silicon x.val=0 value=0 d.extrac assign name=Dj

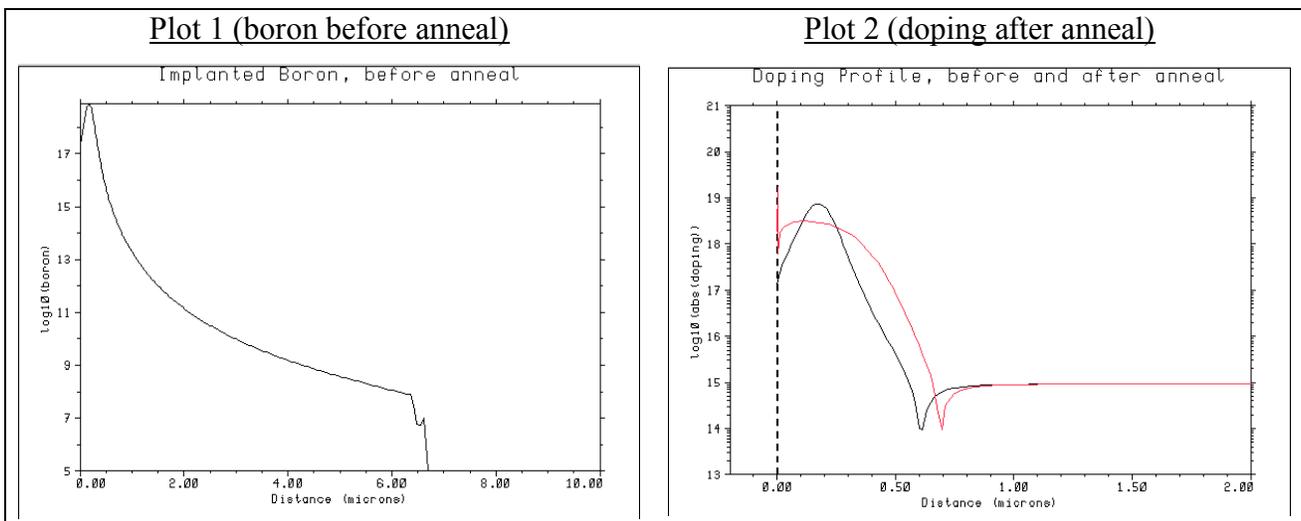
# use abs(doping) to count the carriers N
select z=abs(doping)

# Determine the total carriers in the piezoresistor
# "do a 1-D integration of the active dopants in silicon up to the junction
# depth determined in the previous step"
extract silicon p1.y=0 p2.y=@Dj integral assign name=Ncm2

# Result is given in units of N/cm2, so multiply by area to get
# total N
assign name=N n.val=@Ncm2*@plength*@pwidth print

```

Resulting Plots:



Appendix B

Results of ion implants simulated with Pearson and Monte Carlo models with the input file piezo_ex1.supr. The Monte Carlo method gives different results at different cross-sections. The analytic model gives very small differences which are presumably round-off errors.

Monte Carlo:

```
select      z=doping
print.1d    x.val=0 layers
```

Num	Material	Top	Bottom	Thickness	Integral
1	oxide	-0.0029	0.0007	0.0037	-3.6295e+11
2	silicon	0.0007	0.9387	0.9379	-1.0434e+14
3	silicon	0.9387	10.0000	9.0613	8.2583e+11

```
print.1d    x.val=0.5 layers
```

Num	Material	Top	Bottom	Thickness	Integral
1	oxide	-0.0029	0.0007	0.0037	-3.1683e+11
2	silicon	0.0007	0.9337	0.9329	-9.8917e+13
3	silicon	0.9337	10.0000	9.0663	8.2634e+11

```
print.1d    x.val=1 layers
```

Num	Material	Top	Bottom	Thickness	Integral
1	oxide	-0.0029	0.0007	0.0037	-2.7451e+11
2	silicon	0.0007	0.9292	0.9285	-9.3867e+13
3	silicon	0.9292	10.0000	9.0708	8.2676e+11

Pearson:

```
select      z=doping
print.1d    x.val=0 layers
```

Num	Material	Top	Bottom	Thickness	Integral
1	oxide	-0.0029	0.0007	0.0037	-3.4362e+12
2	silicon	0.0007	0.6899	0.6891	-9.3354e+13
3	silicon	0.6899	10.0000	9.3101	8.4546e+11

```
print.1d    x.val=0.5 layers
```

Num	Material	Top	Bottom	Thickness	Integral
1	oxide	-0.0029	0.0007	0.0037	-3.4348e+12
2	silicon	0.0007	0.6898	0.6891	-9.3293e+13
3	silicon	0.6898	10.0000	9.3102	8.4546e+11

```
print.1d    x.val=1 layers
```

Num	Material	Top	Bottom	Thickness	Integral
1	oxide	-0.0029	0.0007	0.0037	-3.4315e+12
2	silicon	0.0007	0.6898	0.6890	-9.3231e+13
3	silicon	0.6898	10.0000	9.3102	8.4545e+11