2-Mask NMOS TCAD Simulation

The latest TCAD files can be found on GitHub. (Link). Download the link and extract it to your STDB directory or important form SWB.

The 2-mask NMOS process is designed to have the fewest steps in the semiconductor process to get MOSFETS, Diodes, Solar Cells, and Resistors working. The mask set also has features to measure registration errors, as well as an inverter and ring oscillator (Linear, Enhancement Mode NMOS logic.) While the process does work, it is susceptible to alignment errors and process variations. The process is outlined in this paper (Link), from D. W. Parent, "A 2-Mask NMOS Process Design Fabricate and Test Module for Use In Microelectronics Instruction and Process Development," 2006, pp. 57–62. In the paper, enhancement mode devices are created by setting the substrate doping with a born diffusion through a screen oxide. This work uses born implantation.

As things become available, they will be stored in this GitHub repository:
https://github.com/davidparent/2mask-nmos-mpel-sjsu

Notes on the Mask set (Link)

SWB File to download:
https://github.com/davidparent/2mask-nmos-mpel-sjsu/blob/main/TCAD/2mask_nmos_process.gzp

Run everything but the inspect script. Once all the variables have been extracted, you can then run Inspect.

Screening Oxide Step

In the figure below, the SWB shows the first SPROCESS step. It is broken up like this so that when running experiments with the same screening oxide process, you do not have to re-run the same process more than once.

	D	W	NB_INT	TYPE	Grow	SOX_Tmp	SOX_Time
--	5	8.0	8e15	Boron	Grow	880	52

Figure 1: SPROCESS to grow or deposit Screening Oxide.

The screen oxide is grown to protect the wafer's surface and so that the ion during the ion implantation step do not channelize down the spaces between Si atoms.

The simulation assumes a Silicon substrate. The variables are:

- D is the depth of the simulation area in μm.
- W is the width of the simulation area in μm
- NB_INT is the concentration of the boron atoms (Atoms/cm^3).
- TYPE : Set to Boron, the kind of dopant used for the substrate. Other dopants are only supported if you code them in the sprocess file.
- Grow: Grow or any text no equals "Grow." If Grow is activated, the screen oxide is grown in a diffusion step; otherwise, it is just deposited, and a one-minute anneal is conducted to activate the boron in the substrate.
- SOX_Tmp: The temperature at which the oxide will be grown. (Assumes WetOx).
- SOC_Time: The time in minutes the screening oxide will take. This only works if Grow is set to "Grow."

The code is shown below. Notice:

1. How units are set with <>, <um> for microns.
2. Tcl sets variables with set Variable name value
3. Variables from the SWB are read in with @Value@
4. To use a variable from the SWB in Tcl in an if statement, you have to set the SWB variable to a Tcl variable.
5. Tcl variables are references with $varialbe.
6. To print an extracted visible back to the SWB, you use: puts "DOE:
7. Lines 10-15 set the dimensions and the grid.
8. Line 17 set the substrate to Silicon.
9. Line 19 sets the substrate concentration and type.
10. Lines 24-28 set the automatic gris adjustment parameters.
11. Line 32 reads in the Grow variable from the SWB.
12. Lines 34-45 grow or deposit the oxide.
13. Lines 48-56 extract the oxide thickness and print it to the SWB.
14. Line saves the structure. n@node@ will be saved as n17_fps.tdr, if the node is named 17.
15. This file can be opened in svisual and look at doping. It is read in by the next sprocess file.

```
Unset
1. ----------------------------------------------------------------------
```

```
# 2Mask NMOS Example
#-----------------------------------------------------------------------
math coord.ucs

set Depth @D@
set Width @W@
#--- Declare Initial grid ---------------------------------------------
# Note: x=0 is at anticipated top of the epi layer
line x location=0.0<um>   tag=SubTop   spacing= 20.0<nm>
line x location=0.5<um>   spacing= 40.0<nm>
line x location=$Depth<um>   tag=SubBottom   spacing=  0.2<um>

line y location=0.0<um>   tag=SubLeft        spacing= 1.0<um>
line y location=$Width<um>   tag=SubRight    spacing=1.0<um>
#--- Declare substrate ------------------------------------------------
region silicon xlo=SubTop xhi=SubBottom ylo=SubLeft yhi=SubRight
#-- 1-10 Ohm-cm
init concentration=@<NB_INT>@<cm-3> field=@TYPE@

# Global Mesh settings for automatic meshing in newly generated layers
# ---------------------------------------------------------------------

grid set.min.normal.size= 10<nm> set.normal.growth.ratio.2d= 1.5
mgoals accuracy= 1e-5

pdbSet Oxide Grid perp.add.dist 1e-7
pdbSet Grid NativeLayerThickness 1e-7

#---Grow Screening Oxide Layer

set SCREEN @Grow@

if { $SCREEN == "Grow" } {

# Growing screening oxide
# -----------------------
   diffuse temperature=@SOX_Tmp@<C> time=@SOX_Time@<min> H2O
} else {
# Deposit screening oxide
# -------------------------
   deposit material = {Oxide} type = isotropic  rate = {1.0} time=0.1
        # Anneal to activate impurities
   diffuse temperature=@SOX_Tmp@<C> time=@SOX_Time@<min>
}

#--- Measure Screening Oxide
set LAYERS [layers ]
set FirstColumn [list]
foreach Row $LAYERS {
 lappend FirstColumn [lindex $Row 0]
}
set SOX [expr [lindex $FirstColumn 2] - [lindex $FirstColumn 1]]
puts "The thickness of the grown oxide is \ [format %.4f [expr 1e4*$SOX]] A"
puts "DOE: SOX  \ [format %.4f [expr 1e4*$SOX]]"

struct tdr= n@node@

```

```
59. # WritePlx n@node@_gate.plx
```

Figure 2: Tcl Code for process step 1: Grow Screen Oxide.

In the figure below, one can see the Boron concentration distribution after the screening oxide is grown. One can see the boron was gettered into the SiO2. (The colors are added after the fact.) Note how the SiO2 thickness is on both sides of x=0. This is because the process consumes silicon to create SiO_2.

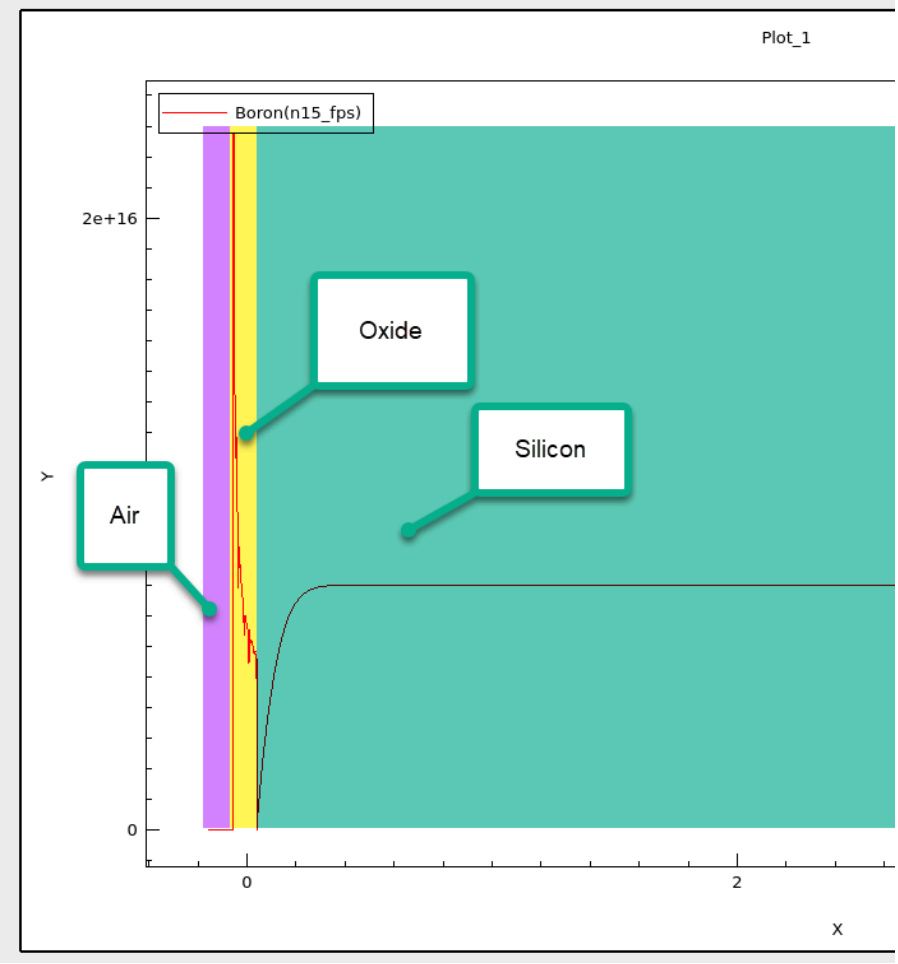


Figure 3: Boron concentration after growing screening oxide. (The top of the wafer is on the left, and the bottom is on the right. Typically, wafers are 100μm thick, but with MOS structures, we try to simulate only the active regions.)

If you scroll in SWB after running the node, you can see the extracted value for screen oxide thickness (SOX)

INSPECT

	SOX
--	1003.5429
--	1003.5429
--	1003.5429

Fix 4: Extracted Screen Oxide Thickness. (Å)

Boron Implantation Step

This step sets the substrate doping concentration (after the well drive and field oxide steps that are shown later) near the Oxide/Si interface. The tilt is set in the sprocess code to 7° to prevent ion channelization. The valuables are:

1. Energy: The energy of the Dose in keV
2. Dose: the dose of the boron, 2-D concretion in Atoms/cm^2.
3. SOX_R: A variable that is set to No (or any but Yes) and controls whether the screen oxide is removed prior to the next processing step.
4. Tilt is assumed to be 7°.

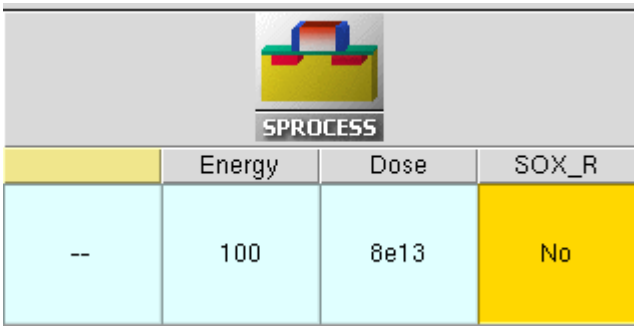


Figure 5: SPROCESS to implant Boron and possibly remove screen oxide.

The sprocess code is shown below.

1. Lines 1-2 read in the output front the previous sprocess node.
2. A Tcl Vailber W is set. (This variable is not used, it was for an extraction when the simulation was forced to be 2-D.)
3. Line 10 is the implantation step.
4. Line 4 is a diffusion step that makes the Boron electrically active.
5. Lines 14-19 extract the substrate doping near the Oxide/Si interface.
6. Lines 21-24 determine if the screen oxide will be removed before the well-drive step.
7. Lines 25 and 26 save the node to be read by the next sprocess simulation.

```
Unset
 1. init tdr=n@previous@
 2. struct smesh=n@node@
 3.
 4. #--- Implant and activate----------------------
 5.
 6. set W @W@
```

```
 7.
 8. # Set the number of steps to show the growth of O2
 9.
10. implant Boron energy=@<Energy>@<keV> dose=@<Dose>@<cm-2> tilt=7<degree>
rotation=0<degree>
11. diffuse temperature=900<C> time=1<min>
12.
13.
14. set Ygox [interface oxide /silicon  ]
15. sel z = { NetActive }
16. set Ygoxtmp [expr $Ygox + 0.005 ]
17. set NA_SOX [format %.3e [lindex [lsort -real [interpolate x = $Ygoxtmp  silicon ]] 0]]
18. set NA_SOX [expr abs($NA_SOX)]
19. puts "DOE: NA_SOX   [format %.3e $NA_SOX]"
20. #--- Choice Remove Screening Oxide
21. set SOX_R @SOX_R@
22. if { $SOX_R == "Yes" } {
23. strip oxide
24. }
25. struct tdr= n@node@
26. fexec rm -rf n@node@_bnd.tdr
27.
```

Figure 6: SPROCESS Tcl Code to implant Boron and possibly remove screen oxide.

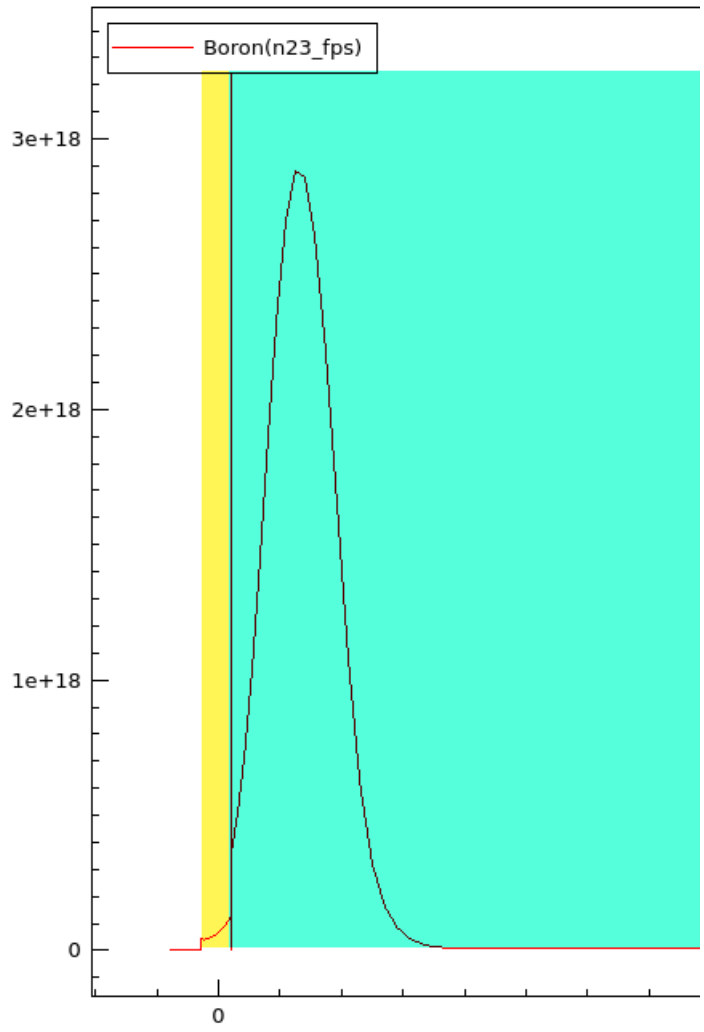


Figure X: SPROCESS Ion implantation results showing the Boron concentration versus X (Depth).

NA_SOX
4.169e+17
4.169e+17
4.169e+17

Fig 7: The extacted substrate doping near the Oxide/Si interface.

Well Drive Step

This step helps set the substrate doping concentration in the region near the Oxide Si interface. The whole wafer is considered the “well”. This is a long step because the same Energey and Dose are used for all NMOS processes in the lab to save money, and the 4-mask process needs this high level of implantation. The variables are:

1. WD_TEMP: The temperature of the well drive step in °C.
2. WD_TIME: The time in minutes of the well-drive step.
3. The gas is not set, so it is assumed that no oxide will grow.

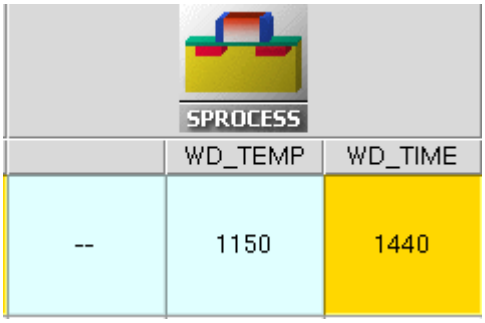


Figure 8: SPROCESS for the well-drive step.

The code:

1. Lines 1-2 read in the previous simulation.
2. Line 6 carries out the well-drive simulation.
3. Lines 6-12 extract the substrate doping near the Oxide/Si Interface.
4. Lines 13-14 save the structure for the next process step.

```
Unset
 1. init tdr=n@previous@
 2. struct smesh=n@node@
 3. #--------------- Well Dirve
 4. set Width @W@
 5.
 6. diffuse temperature=@<WD_TEMP>@<C> time=@WD_TIME@<min>
 7. set Ygox [interface oxide /silicon  ]
 8. sel z = { NetActive }
 9. set Ygoxtmp [expr $Ygox + 0.005 ]
10. set NA_WD [format %.3e [lindex [lsort -real [interpolate   x = $Ygoxtmp  silicon ]] 0]]
11. set NA_WD [expr abs($NA_WD)]
12. puts "DOE: NA_WD   [format %.3e $NA_WD]"
13. struct tdr= n@node@
```

```
14. fexec rm -rf n@node@_bnd.tdr
15.
```

Figure 9: SPROCESS Tcl Code to implement the well-drive step.

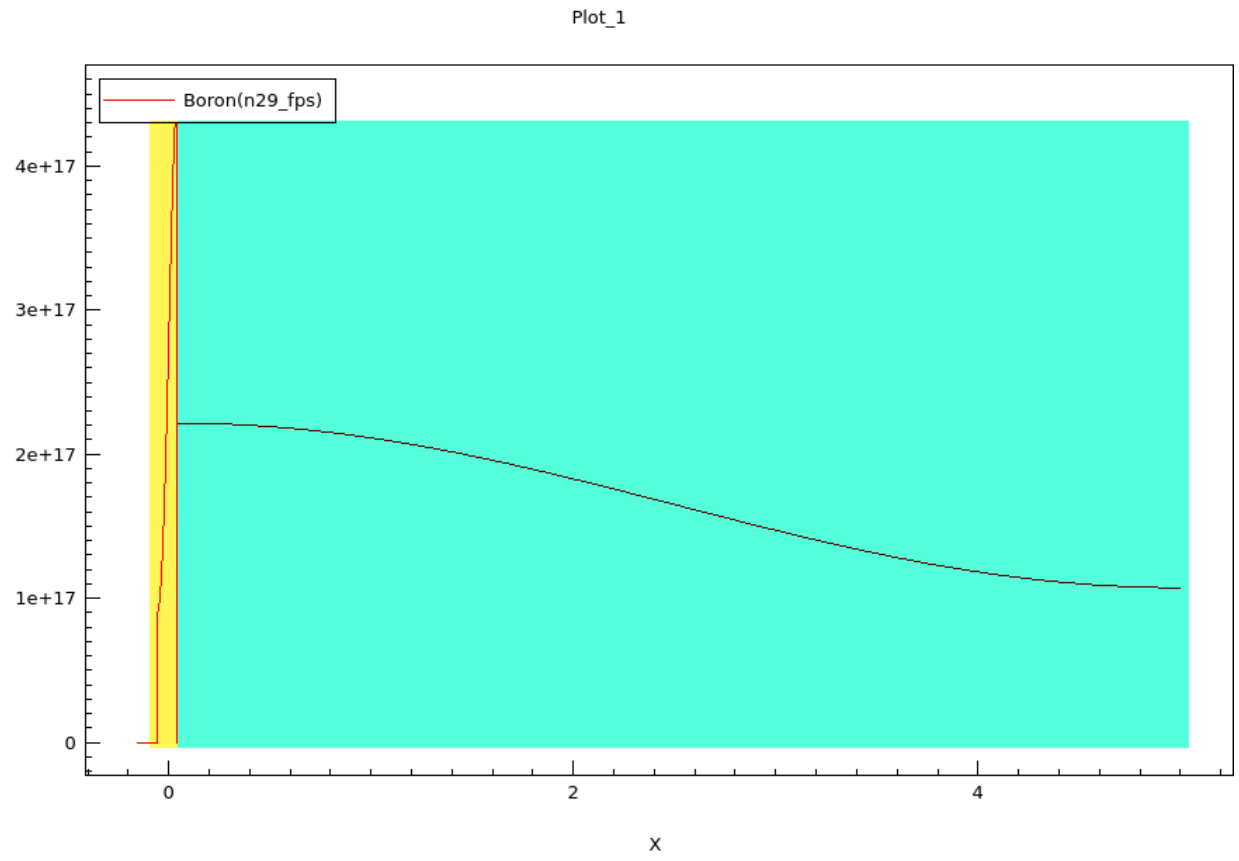


Figure 10: SPROCESS well-drive results. Boron concentration vs Depth. Notice how the Boron diffused in both directions.

NA_WD
2.215e+17
2.215e+17
2.215e+17

Figure 11: Extracted substrate doping near the Oxide/Si interface after the well-drive step.

Field Oxide/Gate Oxide/Diffusion Barrier Growth

This step grows a thick oxide layer with wet oxidation. (Note: The wafers are not taken out of the furnace. The temperature is lowered to the growth temperature and then the oxidation/annealing steps are carried out.) The layer will block phosphorous from diffusing everywhere and only let phosphorous into the source and drains of areas of the MOSFETs and N+ region of the solar cells and the N+ regions of the resistors. This oxide will be etched back to form the gate oxide. The process is wet oxide with HCl, which will prevent light ion contamination (Na+ and K+). There are no separate Field Oxide and Gate Oxide layers, so there will be a large MOS capacitance under every metal line. The variables are:

1. FOX_Tmp: The temperature of the oxidation step in °C.
2. FOX_Time: The time in minutes of the well-drive step.
3. The gas is set to H20 for the oxide growth.
4. The post-oxide anneal is carried out under N2 for 60 minutes at 1100°C. This controls the fixed oxide charge, which makes NMOS threshold voltages more negative. “Modern” processes use a 2-D ion-implantation step to set the threshold voltage.

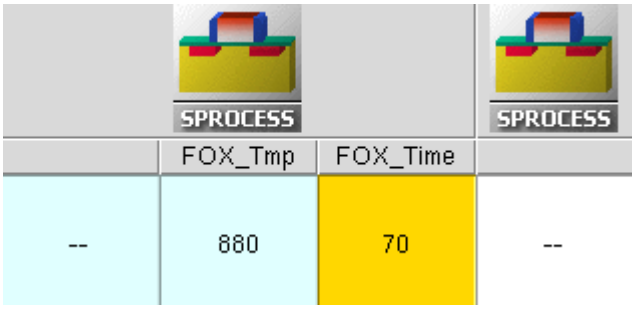


Figure 12: SPROCESS Step to grow the Field Oxide/Gate Oxide/Diffusion Barrier.

```
Unset
  1. init tdr=n@previous@
 2. struct smesh=n@node@
 3. set Width @W@
 4. # ---------------Grow Field Oxide Wet
 5. diffuse temperature=@<FOX_Tmp>@<C> time=@FOX_Time@<min> H2O
 6. #-- Anneal
 7. diffuse temperature=1100<C> time=60<min>
 8. set LAYERS [layers  ]
 9. set FirstColumn [list]
10. foreach Row $LAYERS {
```

```
 lappend FirstColumn [lindex $Row 0]
}
set FOX [expr [lindex $FirstColumn 2] - [lindex $FirstColumn 1]]
puts "The thickness of the grown oxide is \ [format %.4f [expr 1e4*$FOX]] A"
puts "DOE: FOX  \ [format %.4f [expr 1e4*$FOX]]"
set Ygox [interface oxide /silicon   ]
sel z = { NetActive }
set Ygoxtmp [expr $Ygox + 0.005 ]
set NA_FOX [format %.3e [lindex [lsort -real [interpolate  x = $Ygoxtmp  silicon ]] 0]]
set NA_FOX [expr abs($NA_FOX)]
puts "DOE: NA_FOX   [format %.3e $NA_FOX]"
struct tdr= n@node@
fexec rm -rf n@node@_bnd.tdr
```

Figure 13: SPROCESS Tcl Code to implement the Field/Gate/Diffusion Barrier step.

FOX	NA_FOX
2131.2994	1.825e+17
2131.2994	1.825e+17
2131.2994	1.825e+17

Figure 14: Extracted Field Oxide Thickness (Å) and substrate doping (cm^{-3}) after the oxidation and anneal step.

One can use these values to estimate the 1-D threshold voltage for a fixed oxide charge of $5 \times 10^{11} q/cm^2$ with an aluminum gate. (Co-Lab Code). The threshold voltage is near 8 V.

MASK 1: NDIFF

So far, the situation has all been 1-D. Once we do a masking, etch and photoresist removal steps, the simulation will be 2-D automatically. The simulation steps will now proceed more slowly. Fortunately, for this 2-mask nmos process, there are no more time-intensive diffusion steps. While one could add variables for this process, the process steps are fixed. There are no variables taken from the SWB. At the end of this step, a window will be etched in the S/D area. Note to save on simulation resources: we only simulate half of the device for now.

Figure 15: SPROCESS for Mask 1:NDIFF (N-type Difusion)

The Code:

1. Lines 1-2 read in the previous structure.
2. Lines 4-5 are an experiment at stretching the structure. They are not used.
3. Line 7 creates a amsk from 0 to 6 microns that is negative. This means windows in the photo resist will be opened form 0 to 5 microns from left to right.
4. Line 8 sets the photoresist thickness and opens the NIFF region
5. Line 9 saves the structure for plotting.
6. Line 10 etches the oxide. This time should be changed to the extracted thickness times 1.1 divided by the rate. As it is now, it will try to etch 3000Å. Since the filed oxide from this example is 2131Å, we are over-etching by more than 10%/. Also, if one grows a Field oxide greater than 3000Å, the source and drain area would not be adequately etched.
7. Line 11 saves the structure for plotting.
8. Line 14 strips the photoresist.
9. Lines 16-17 save the structure for the next node.
10. One could extract the width of the etched area to explore over etching's effect on the channel length.

```
Unset
 1. init tdr=n@previous@
 2. struct smesh=n@node@
 3.
 4. #transform stretch Adaptive location=1.0<um> right length=7<um>
 5. #transform stretch Adaptive location=1.0<um> down length=2<um>
 6.
 7. mask name=mask1_ndiff segments= {0 5} !negative
 8. photo mask=mask1_ndiff thickness=1<um>
 9. struct tdr= n@node@_mask1_ndiff
10. etch material= {Oxide}         type= isotropic time= .3 rate= {1}
11. struct tdr= n@node@_mask1_ndiff_after_etch
12.
13. # strip oxide
14. strip photoresist
15.
16. struct tdr= n@node@
17. fexec rm -rf n@node@_bnd.tdr
```

Figure 16: SPROCESS Tcl code for mask 1 and oxide etch.

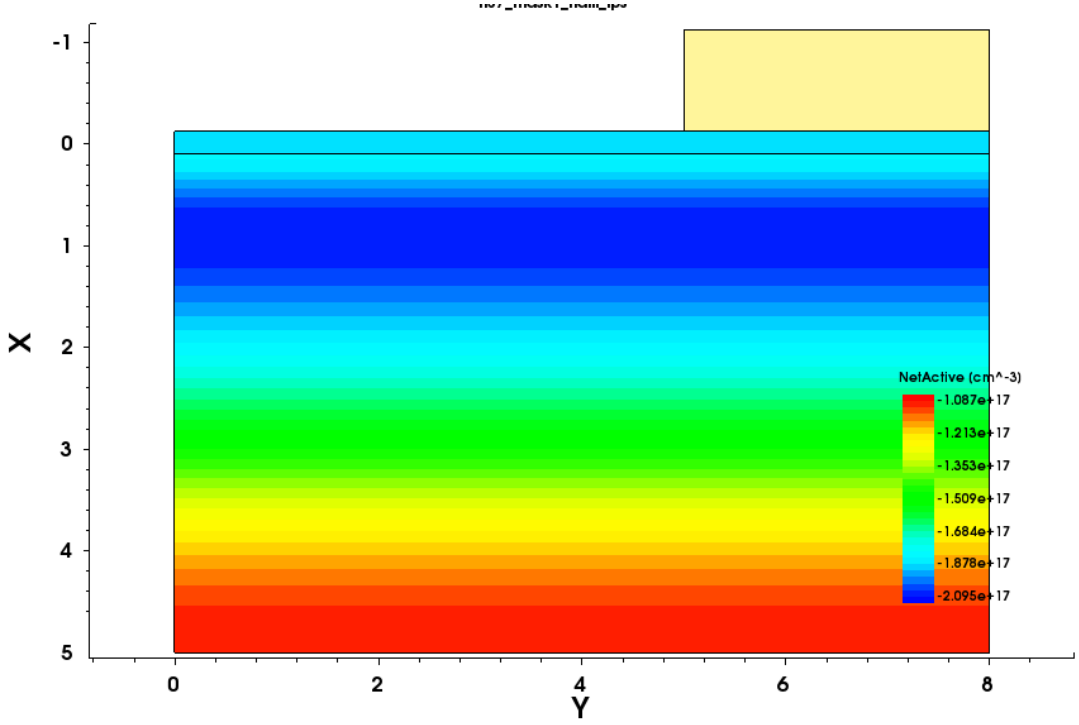


Figure 17: Results of SPROCESS MASK1

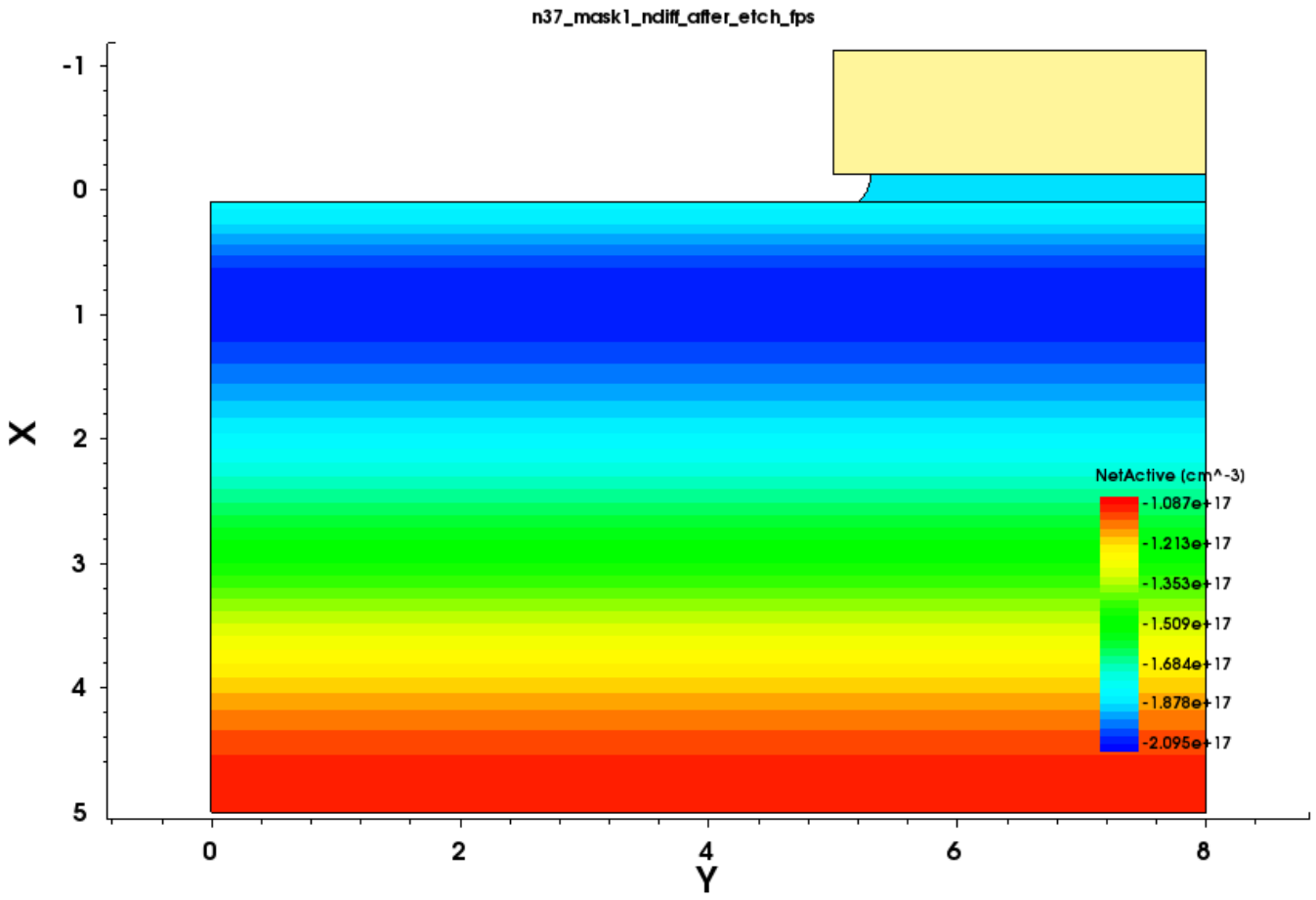


Figure 18: Results of SPROCSS Oxide etch step. Notice undercutting. This is due to using wet comical etching, rather than RIE.

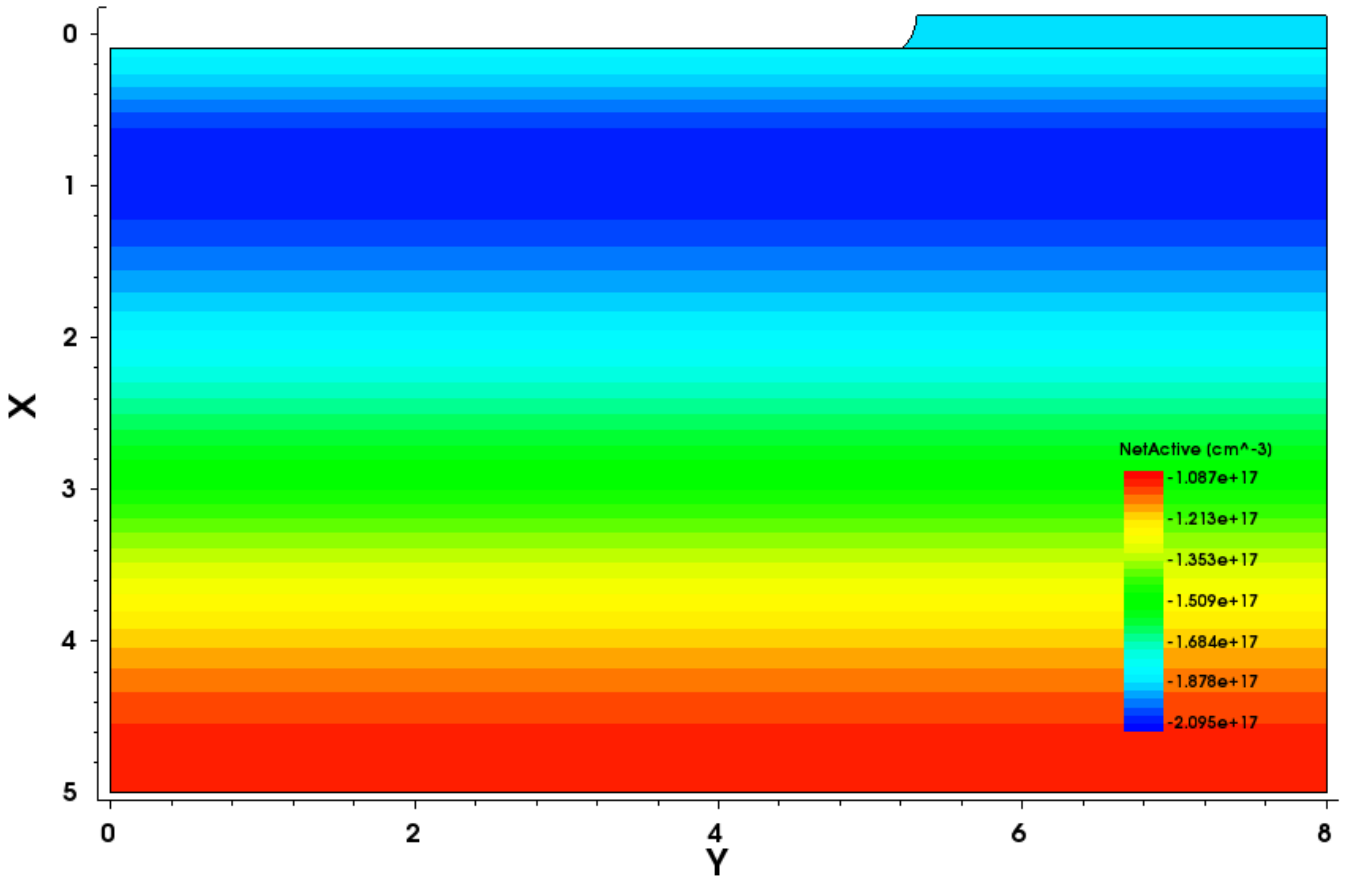


Figure 19: Results of removing the photoresist.

NDIFF Diffusion step

This step deposited a spin on glass with a phosphorus dopant source. In reality, we use a solid source, but at this point, I need help figuring out how to simulate it. One the SOG s spun on, the wafers are diffused and the s/D area are not doped. Since this is a high-temperature step, the dopants move around, so we need to extract the substrate doping again. The final step to etch the SOG (oxide grows with the sold source as well, so this needs to be done.) We over-etch the oxide to a smaller VT. It is difficult to control the threshold voltage in this manner, so this would not be viable for a commercial process, but for workforce development and training, it is acceptable. The variables:

1. diff_temp : This is the diffusion temperature. (°C)
2. diff_time : The time in minutes of the diffusion step/
3. etch_rate: The BOE etch rate microns/mimute.
4. Etch_time: the boe etch time. This will etch away the SOG exactly, an the SOG was departed at a rate of 1 for .3 minutes/

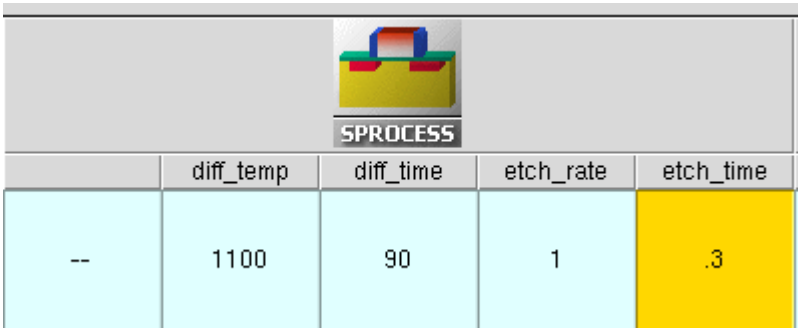


Figure 20: SPROCESS SOG deposition, n-type diffusion step, SOG removal, and Field Oxide thinning to Gate Oxide.

The code:

1. Lines 1-2 are read in the previous step.
2. Lines 5-6 deposit the SOG and save the file for plotting.
3. Lines 9-10 diffuse the dopant and save the structure for plotting.
4. Lines 12-13, etch the SOG, set the Gate oxide thickness, and save the file for plotting.
5. Lines 15-16 save the structure for the next node.

```
Unset
 1. init tdr=n@previous@
 2. struct smesh=n@node@
 3.
 4. #--- Deposite SOD-----------------------
 5. deposit material = {Oxide} type = anisotropic  Phosphorus conc=5e21 rate = {1.0} time=0.3
 6. struct tdr= n@node@_sod_dep
 7.
 8. #--- Pre Dep ------------------------------------------------
 9. diffuse temperature=@<diff_temp>@<C> time=@<diff_time>@<min>
10. struct tdr= n@node@_ndiff
11. # Etch SOD plus some field
12. etch material= {Oxide}         type= anisotropic time= @<etch_time>@ rate= @<etch_rate>@
13. struct tdr= n@node@_sod_etch
14.
15. struct tdr= n@node@
16. fexec rm -rf n@node@_bnd.tdr
```

Figure 21: SPROCESS Tcl code for deposition, n-type diffusion step, SOG removal, and Field Oxide thinning to Gate Oxide.

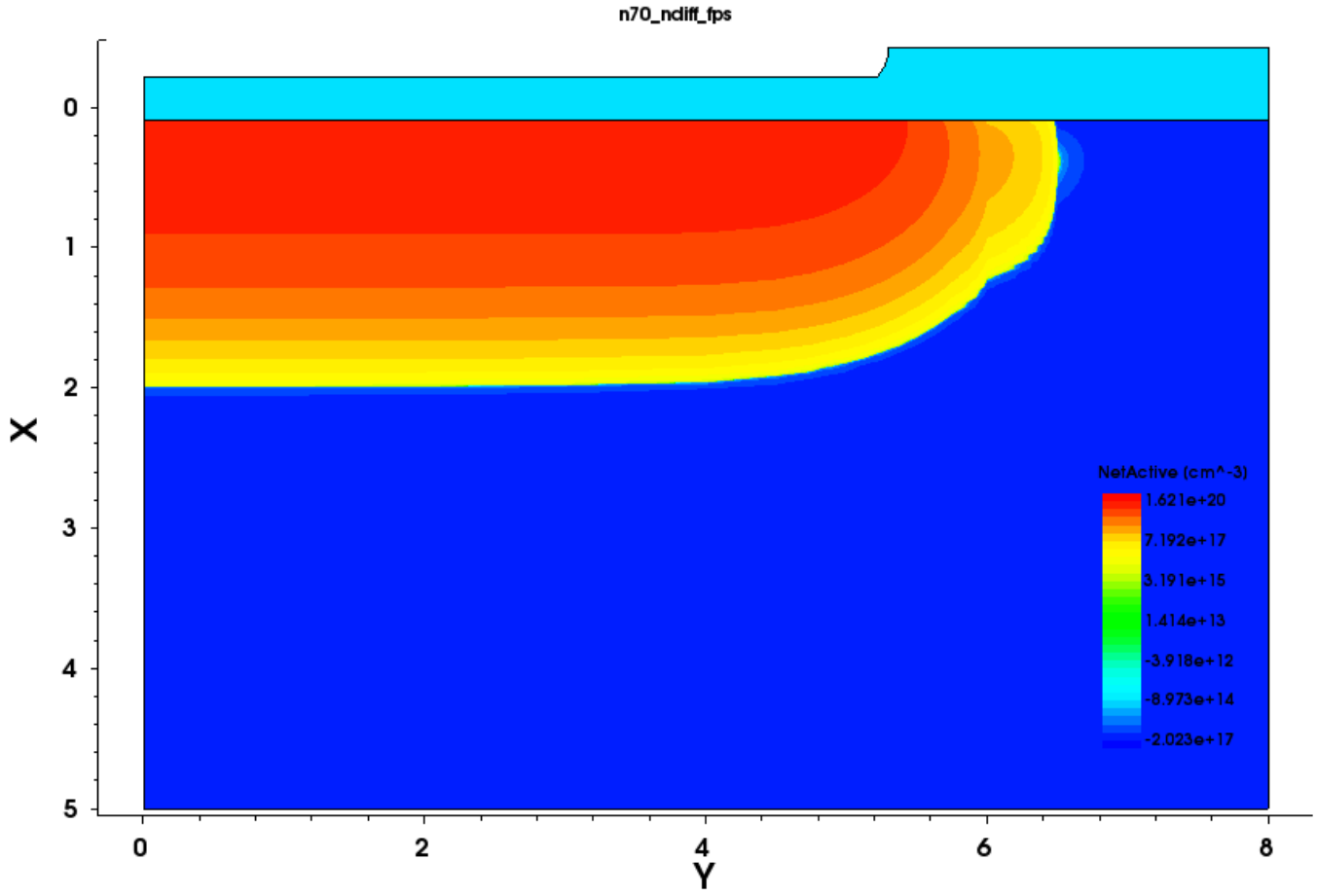


Figure 22: 3000Å of Spin on Glass (SOG), sometimes called Spin on Dopant (SOD) deposited on the wafers after the diffusion step.

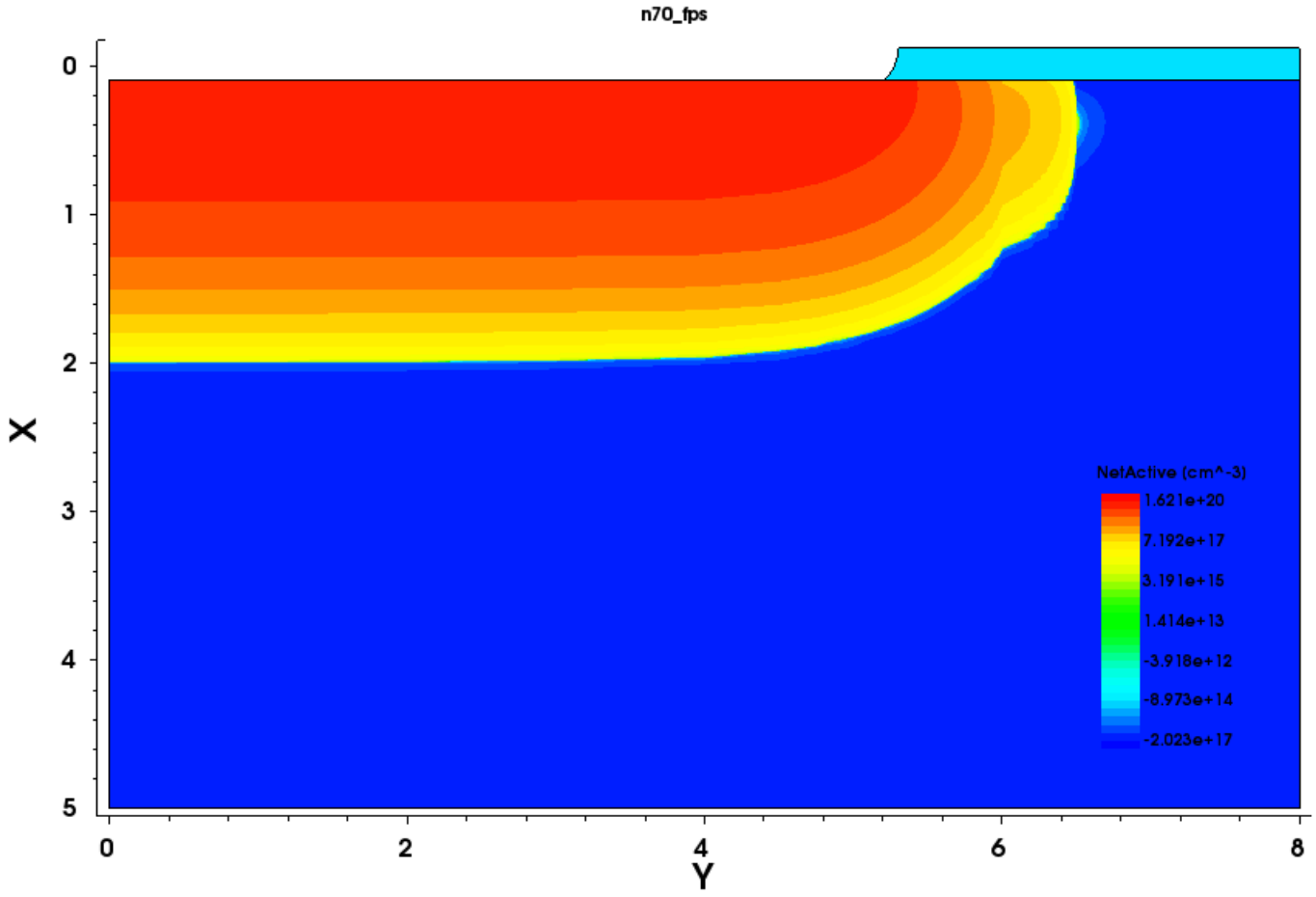


Figure 23: Results of diffusion step and removal of SOG. In this case, only the SOG was removed. To thin the gate oxide, increase the etch time.

Aluminum Deposition and Mask 2

This step simulates the deposition of Alumnim onto the wafer, the application of photoresist, the use of mask 2 (METL 1) to expose the desired pattern, etching, and photoresist removal.

The offset variable "off", is used to simulate the registration error. The simulation does not match reality in the gate metal is moved in along the y-direction, so one would get an asymmetrical device. On one extreme, the left side of the gate metal would not overlap the S/D (still separated by oxide) and so there would be a region that was not inverted, so the device would be harder to turn on, while the right side would have the aluminum short the gate to the S/D region. The case just switches the left and right-hand sides. A short is easy to see under the microscope, and so we just look at when the gate can not invert the surface from the source to the drain. This can appear as a large series resistance between the probe and the source and drain regions, thus increasing the effective threshold voltage. The effect is more pronounced the thinner the gate oxide. Simulations show that a 1.5 μm offset dramatically increases the turn-on voltage.

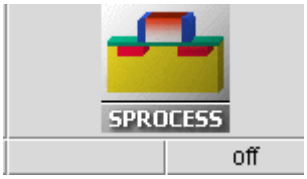


Figure 24: SPROCESS for aluminum deposition and Mask 2 patterning.

The code:

1. Lines 1-2 read in the previous structure.
2. Line 4 deposits the aluminum.
3. Line 5 saves the structure for plotting.
4. Line 7 sets the mask edge (right most area that will be etched plus the offset.)
5. Lines 9, 10, and 11 pattern the photoresist for mask 2 and save it for plotting.
6. Lines 13-14 etch the Aluminium and save the structure for plotting.
7. Line 16 strips all the photoresist.
8. Lines 17-35 extracts the oxide thickness and substrate doping.
9. Lines 36 and 37 set a variable Ymet so that the gate contact can be automatically placed, regardless of how thick the gate oxide is.
10. Lines 39-44 set the Source, Gate, and substrate contacts and sames the structure for mirroring.
11. Lines 46-48 mirror the struct, add the drain contact and save for simulation on the next node.

Unset

```
init tdr=n@previous@
struct smesh=n@node@

deposit material = {Aluminum} type = anisotropic  rate = {3.0} time=0.124
struct tdr= n@node@_m1_dep

set mask_edge [expr 5.0 + @<off>@]

mask name=mask2_aluminum segments= {3 $mask_edge} !negative
photo mask=mask2_aluminum thickness=1<um>
struct tdr= n@node@_m2

etch material= {Aluminum}      type= isotropic time= .134 rate= {3}
struct tdr= n@node@_m2_etch

strip photoresist
set Ygox [interface oxide /silicon y=8.0<um>]
select z=1
set LAYERS [layers y = 8.0<um> ]
set FirstColumn [list]
list
foreach Row $LAYERS {
 lappend FirstColumn [lindex $Row 1]
}
set TOX [expr [lindex $FirstColumn 2] - [lindex $FirstColumn 1]]
puts "The thickness of the grown oxide is \ [format %.4f [expr 1e4*$TOX]] A"
puts "DOE: TOX  \ [format %.4f [expr 1e4*$TOX]]"

set Ymet [interface Aluminum /oxide y=8.0<um>]
set Ymet [expr $Ymet -0.05 ]
sel z = { NetActive }
set Ygoxtmp [expr $Ygox + 0.005 ]
set NA_TOX [format %.3e [lindex [lsort -real [interpolate y=8.0<um> x = $Ygoxtmp  silicon
]] 0]]
set NA_TOX [expr abs($NA_TOX)]
puts "DOE: NA_TOX   [format %.3e $NA_TOX]"
set Ymet [interface Aluminum /oxide  y=8.0<um>]
set Ymet [expr $Ymet -.05]

contact name= substrate bottom
contact name= source point y=.1       x= 0.0 !replace

contact name= gate   point y= 8       x= $Ymet !replace

struct tdr= n@node@

fexec rm -rf n@node@_bnd.tdr
fexec tdx -mtt -Y -ren "source=drain" n@node@_fps.tdr n@node@_ref.tdr
fexec mv -f n@node@_ref.tdr n@node@_fps.tdr
```

Figure 25: SPROCSS Aluminum layer deposited on the wafers Tcl Code

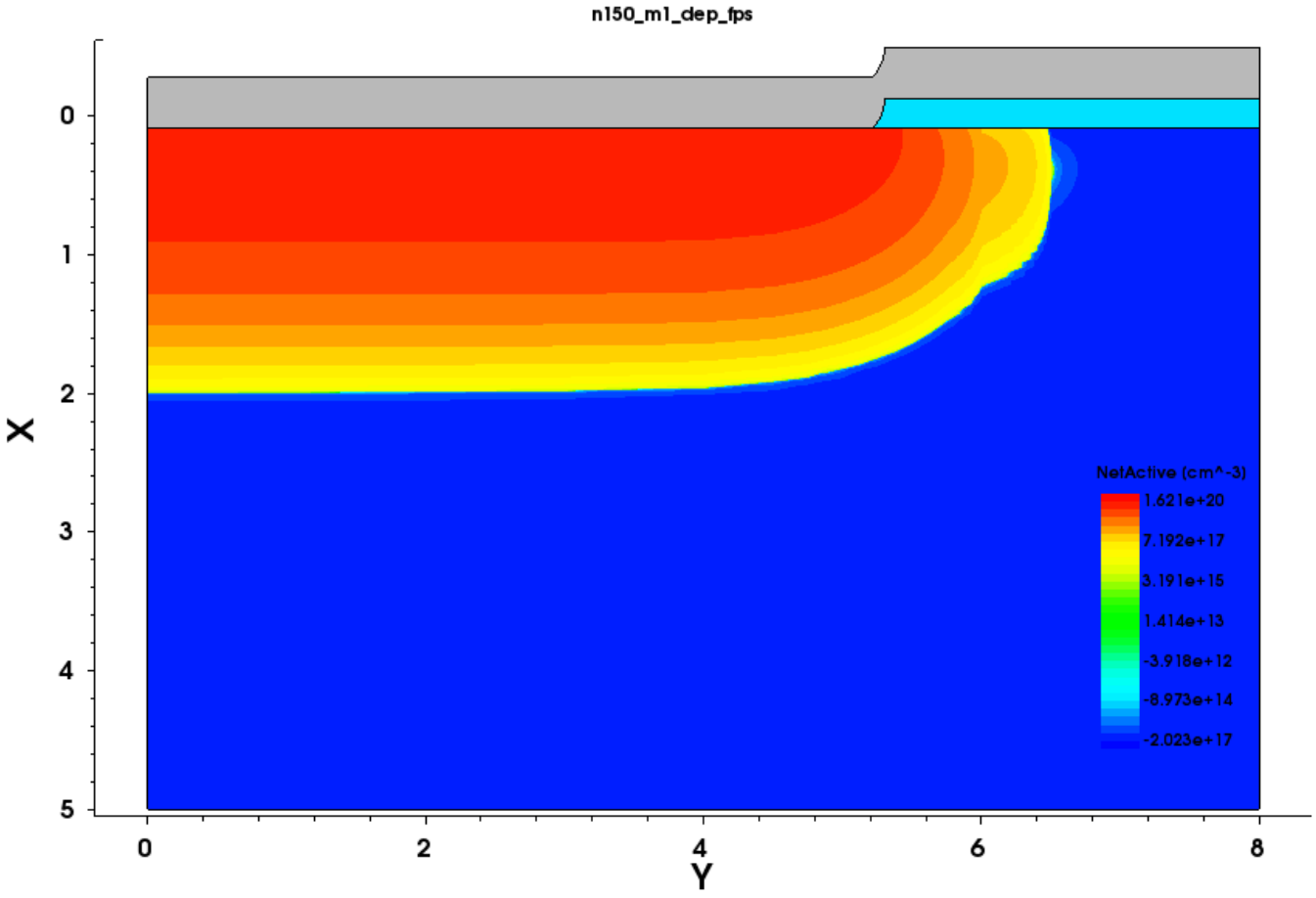


Figure 26: SPROCSS Aluminum layer deposited on the wafers.

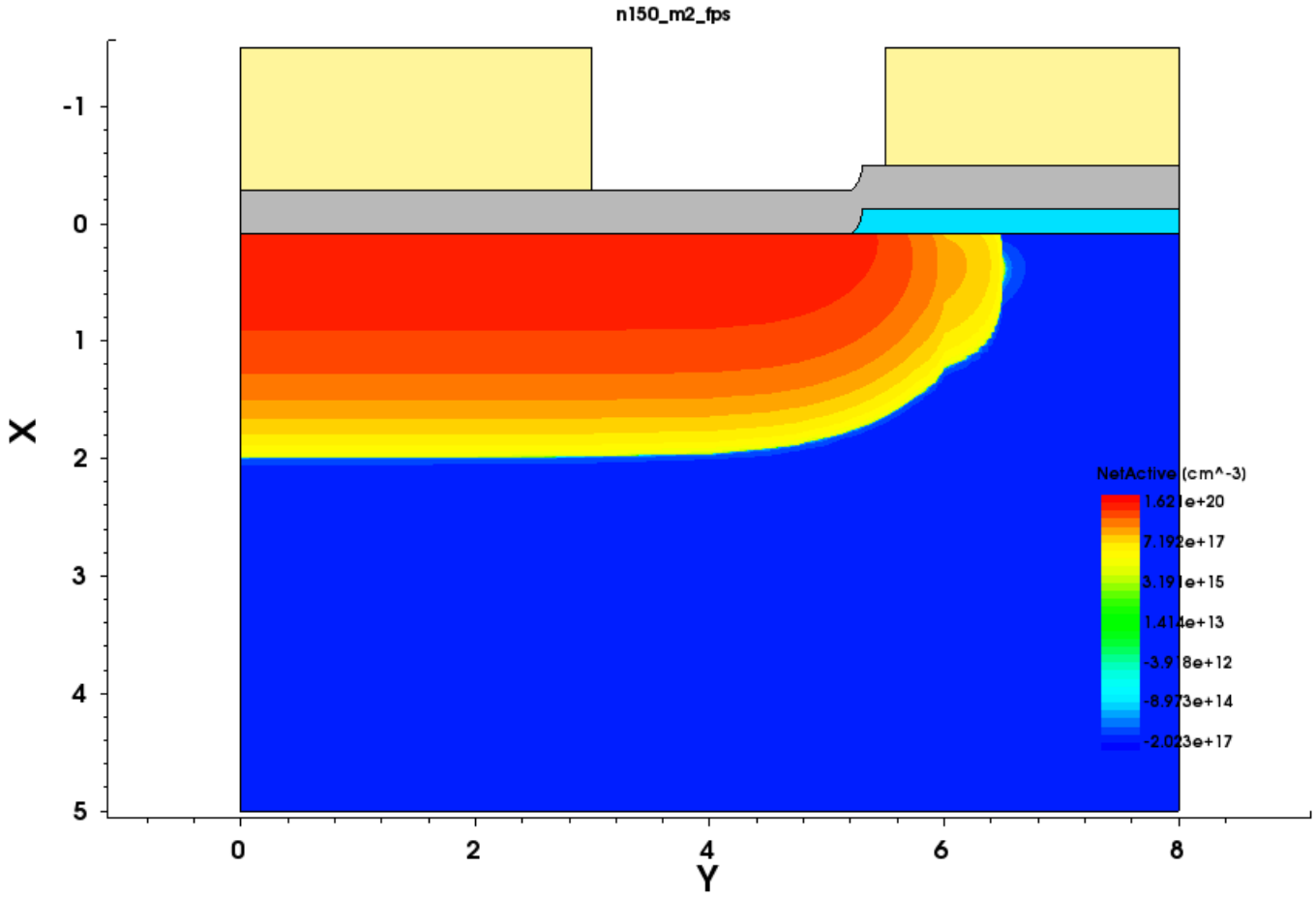


Figure 27: SPROCSS Photo resist deposited, exposed with mask 2, and developed. Ready for Al etch step.

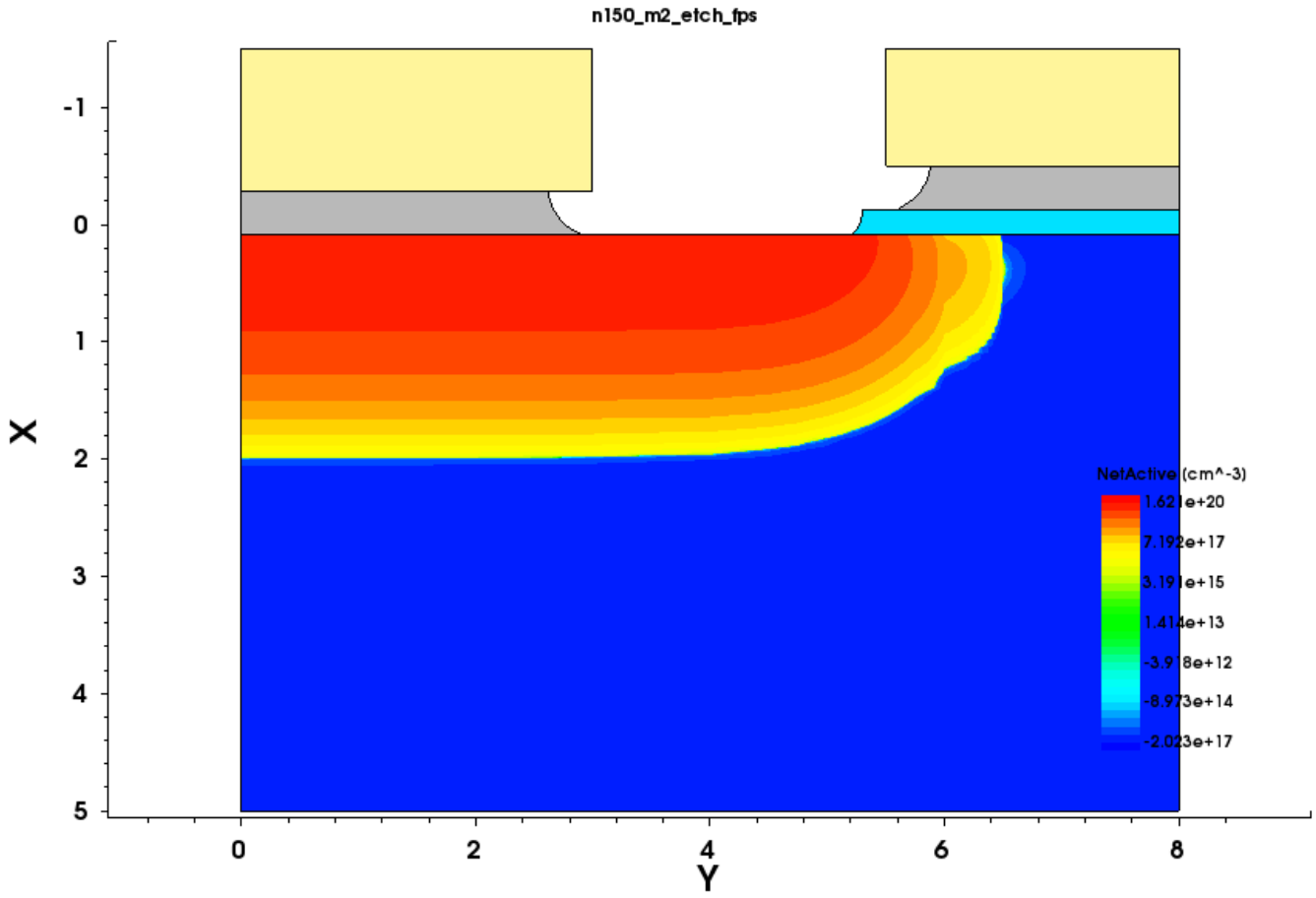


Figure 28: SPROCSS separation of S/D from gate, Aluminum etched.

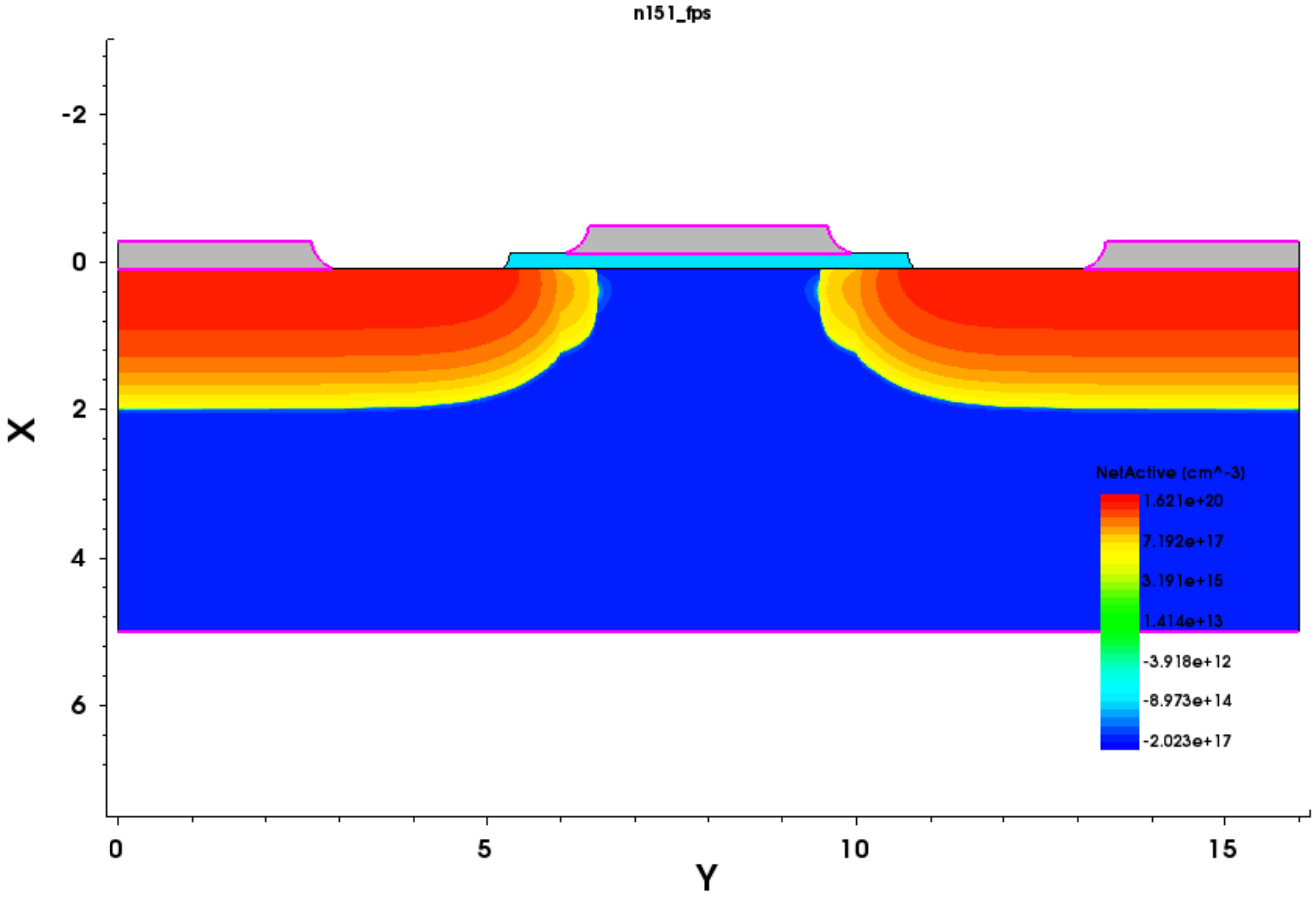


Figure 29: Photoresist removed, mirror around the x-axis. Contacts were created, and the device is ready for electrical simulation. The offset variable is set to 0.5 μm to simulate a 0.5 μm registration error in the y direction.

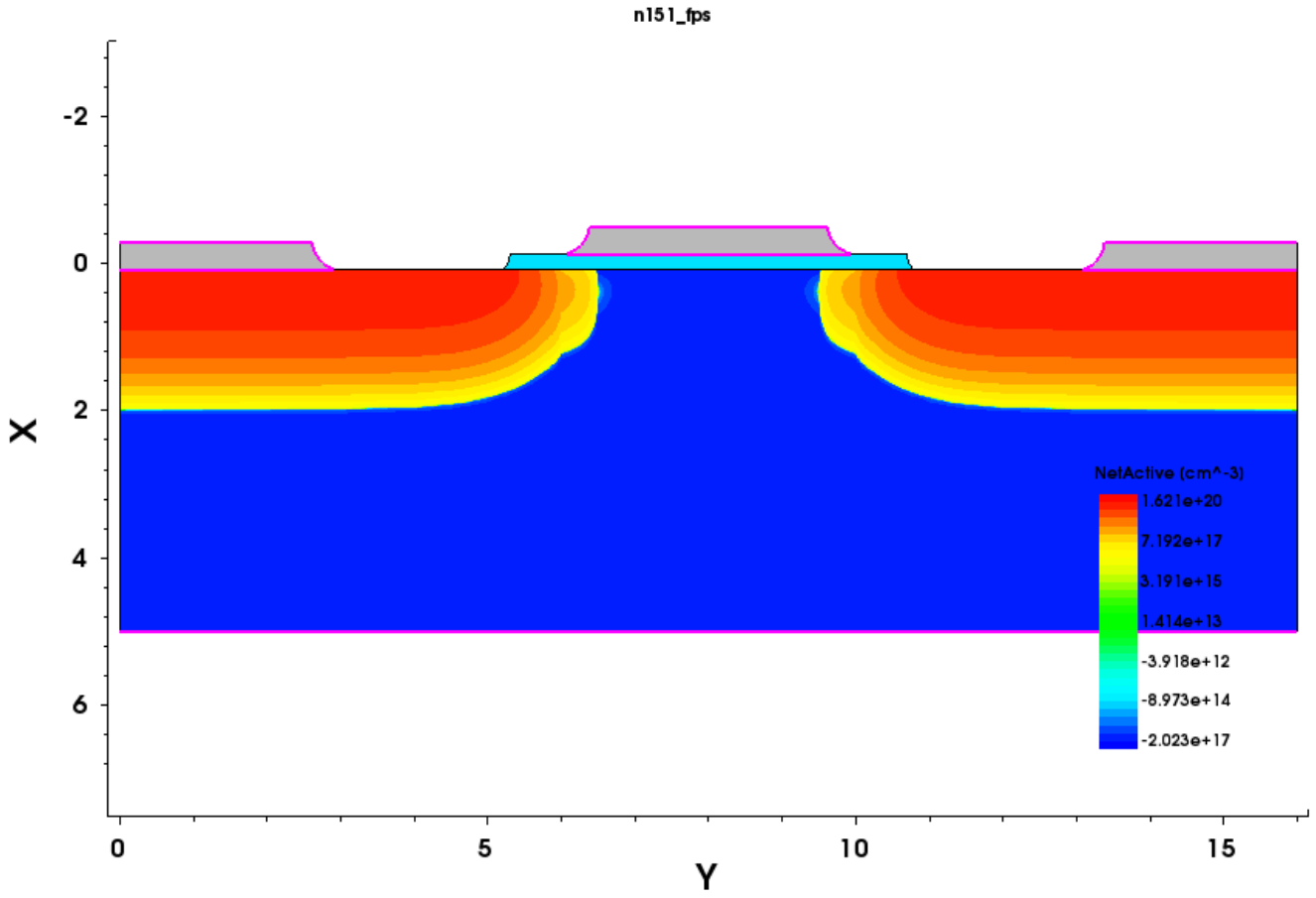


Figure 30: Photoresist removed, mirror around the x-axis. Contacts were created, and the device is ready for electrical simulation. The offset variable is set to 1 μm to simulate a 1 μm registration error in the y direction.

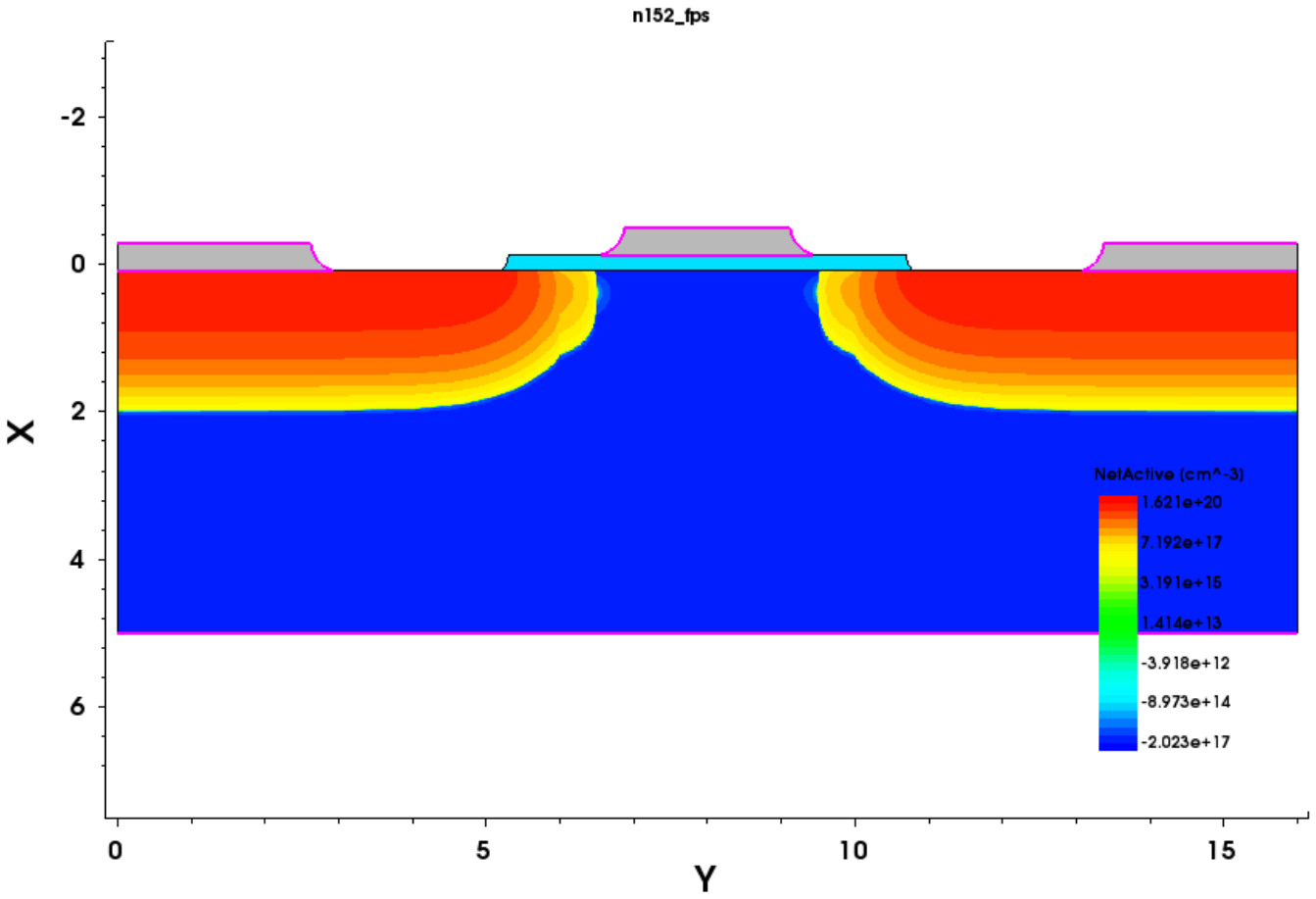


Figure 31: Photoresist removed, mirrored around the x-axis. Contacts were created, and the device is ready for electrical simulation. The offset variable is set to 1.5 μm to simulate a 1.5 μm registration error in the y direction. Notice that the edges of the gate do not overlay the S/D regions.

Electrical Stimulation of the MOSFET (Linear mode VT extraction)

Variables:

1. Vds The drain-source voltage of the simulation
2. Vdd, The gate-source voltage the that the gate is swept to.
3. QSS, Fixed oxide charge (Not really charge, just defecfts per cm^{-2}.)

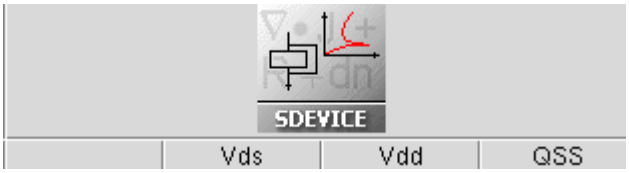
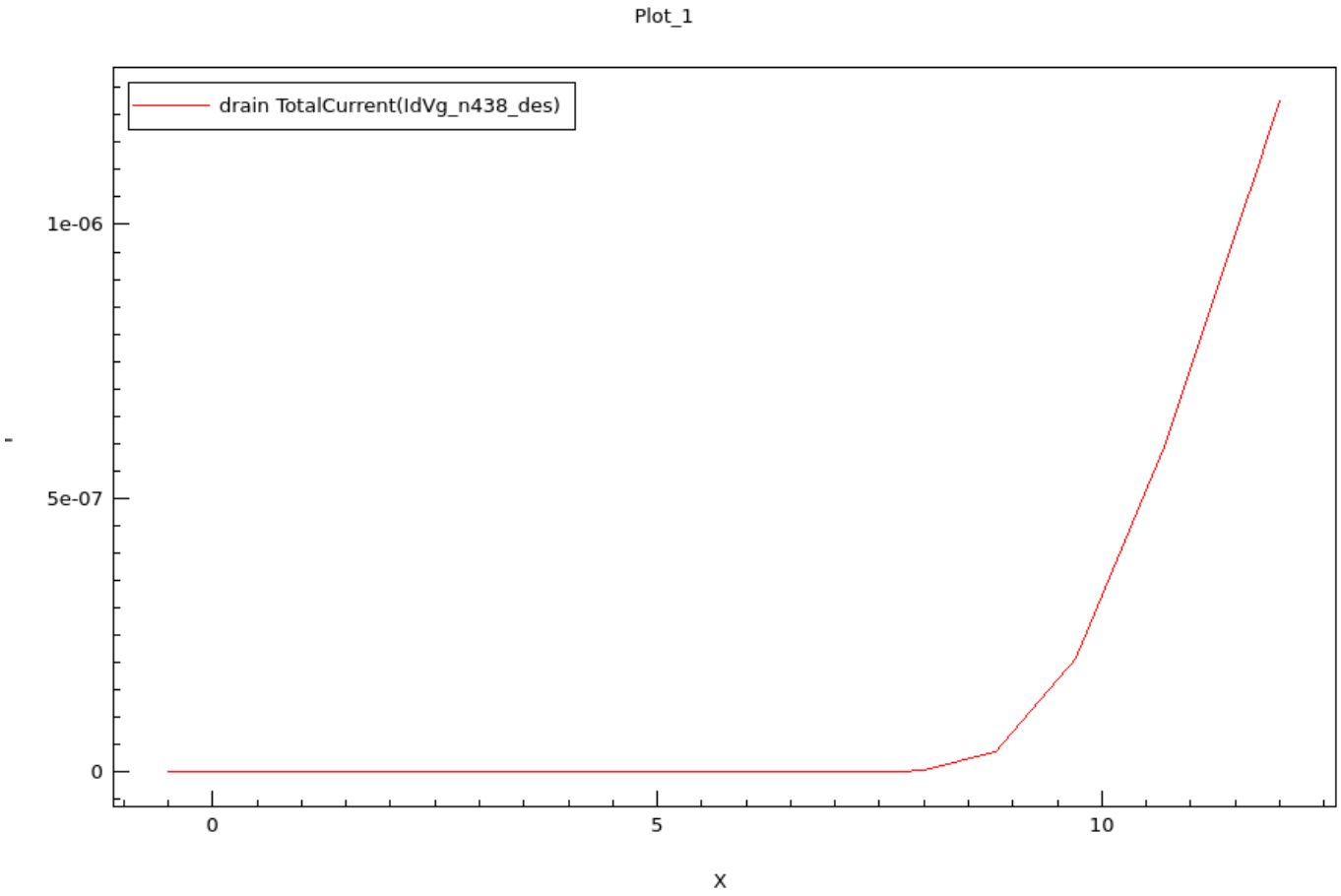


Figure 32: SDEVICE Node

```
Unset
( 
  set SIGN   1.0
  set EQNS   "Poisson Electron"
)!

File {
    * input files:
    Grid=   "@tdr@"
    Parameter="@parameter@"
    * output files:
    Plot=   "@tdrdat@"
    Current="@plot@"
    Output= "@log@"
}

Electrode {
    { Name="source"      Voltage=0.0   }
    { Name="drain"       Voltage=@Vds@   }
    { Name="gate"        Voltage=0.0  Material=Aluminum}
    { Name="substrate" Voltage=0.0 }
}

```

```
Physics{
   EffectiveIntrinsicDensity( OldSlotboom )
}
Physics( MaterialInterface="Silicon/Oxide") { charge(Conc=@QSS@) }

Physics(Material="Silicon"){

   Mobility(
      PhuMob
      HighFieldSaturation
      EnormalHigh
   )
   Recombination(
      SRH( DopingDep )
   )
}

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

Solve {
*- Creating initial guess:
   Coupled(Iterations=100 LineSearchDamping=1e-4){ Poisson }
   Coupled { !(puts $EQNS)! }


*- Ramp to VG to -2
   Quasistationary(
      InitialStep=1e-1 Increment=1.35
      MinStep=1e-6 MaxStep=0.5
      Goal { Name="gate" Voltage=-.50 }
   ){ Coupled { !(puts $EQNS)! } }
*- Vg sweep
   NewCurrentFile="IdVg_"
   Quasistationary(
      DoZero
      InitialStep=1e-5 Increment=1.1
      MinStep=1e-5 MaxStep=0.25
      Goal { Name="gate" Voltage=!(puts [expr $SIGN*@Vdd@])! }
   ){ Coupled { !(puts $EQNS)! }
      CurrentPlot( Time=(Range=(0 1) )  )
   }
}
```

Figure 33: Sdevice code.



Fgiure 34: ID vs VGS, with VDS held contant. Use to extract VT, gm, subthreshold slope.

Extraction

Figure 35: Inspect tool. Should switch to sviusal at some point.

```
Unset
1. set NN    @node@
2. set i     @node:index@
3. set Vds   @Vds@
4. set Vg    @Vdd@
5. set tox @TOX@
6. set na @NA_TOX@
```

```
set qi @QSS@
set ID    "nMOS"
set Type "nMOS"


#- Automatic alternating color assignment tied to node index
#-------------------------------------------------------------------------#
set COLORS  [list green blue red orange magenta violet brown]
set NCOLORS [llength $COLORS]
set color   [lindex  $COLORS [expr $i%$NCOLORS]]

#- INSPECT IdVg plotting
#-------------------------------------------------------------------------#
# Plotting Id vs Vg curves
gr_setTitleAttr "IdVg  Vds=$Vds"

proj_load  IdVg_@plot@ PLT($NN)

cv_createDS IdVg($NN) \
 "PLT($NN) gate OuterVoltage" "PLT($NN) drain TotalCurrent" y
cv_abs IdVg($NN) y
cv_setCurveAttr IdVg($NN) "IdVg $ID" \
  $color solid 2 none 3 defcolor 1 defcolor

gr_setAxisAttr X  {Gate Voltage (V)}  16  {} {} black 1 14 0 5 0
gr_setAxisAttr Y  {Drain Current (A/um)} 16  {} {} black 1 14 0 5 1

cv_createWithFormula curve_4 "diff(<IdVg($NN)>)" A A

cv_display curve_4 y
cv_createWithFormula curve_5 "<curve_4>/<IdVg($NN)>" A A
cv_display curve_5 y
set max_gm_id   [ cv_compute "vecmax(<curve_5>)"  A A A A ]
set N                        [ expr 1.0/0.0259/$max_gm_id  ]



#script_sleep 5



#- Extraction
#-------------------------------------------------------------------------#
#source EXTRACT_ins.lib

#- Defining current level for Vti extraction
#-------------------------------------------------------------------------#
load_library EXTRACT
set Vtgm [ExtractVtgm Vtgm IdVg($NN) nMOS]

```

```
57. if { $Vds < 0.5 } {
58. set Vtb   [ExtractVtgmb Vtgm  IdVg($NN) $Type]
59. }
60. set xmin [expr 0.55*$Vtgm]
61. set max_gm_id   [ cv_compute "vecmax(<curve_5>)"  $xmin A A A ]
62. set N                        [ expr 1.0/0.0259/$max_gm_id  ]
63.
64.
65. set Idmax [ExtractMax   Id     IdVg($NN)]
66. set SS    [ExtractSS    SS     IdVg($NN) $xmin]
67. set gm    [ExtractGmb   gm     IdVg($NN) $Type ]
68. puts "DOE: max_gm_id [format %.2f $max_gm_id] "
69.
70. puts "DOE: N [format %.2f $N] "
71. set q 1.6e-19
72. set tox [expr $tox * 1e-8]
73. set na $na
74. set ni 9.65e9
75. set t 300
76. set k 1.38e-23
77. set phi_p [expr $k * $t / $q * log( $na / $ni)]
78. set phi_ms [expr -0.6 - $phi_p]
79. puts "PHI_MS= $phi_ms"
80. set qi [expr $qi* $q ]
81. set eps_si [expr 8.85e-14 * 11.9 ]
82. set qd [expr -2 * sqrt($eps_si * $na * $phi_p * $q) ]
83. puts "QD= $qd"
84. set cox [expr 3.9*8.85e-14 / $tox]
85. puts "Cox = $cox"
86. set VT_LC [expr $phi_ms - $qi / $cox - $qd / $cox + 2* $phi_p ]
87. puts "VT Long Channel = $VT_LC"
88. puts "DOE: VT_LC [format %.2f $VT_LC] "
89.
```

Figure 36: Inspect script.

SOX	NA_SOX	NA_WD	FOX	NA_FOX	TOX	NA_TOX	Vtgm	Id	SS	gm	max_gm_id	N	VT_LC
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	2131.2994	1.980e+17	9.531	1.227e-06	766.824	4.968e-07	3.68	10.50	9.78
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	2131.2994	1.980e+17	11.719	5.802e-10	69099.937	2.068e-09	3.76	10.27	14.23
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	2131.2994	1.980e+17	9.599	1.299e-06	769.196	5.410e-07	3.67	10.51	9.78
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	2131.2994	1.980e+17	11.709	5.833e-10	3774.813	2.001e-09	3.76	10.27	14.23
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	2131.2994	1.980e+17	9.291	1.032e-06	771.785	3.813e-07	3.69	10.46	9.78
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	2131.2994	1.980e+17	10.879	6.830e-11	20759.654	6.092e-11	2.96	13.03	14.23
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	1131.2994	1.980e+17	4.822	6.579e-06	406.917	9.311e-07	6.41	6.03	5.12
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	1131.2994	1.980e+17	7.184	4.452e-06	540.626	9.310e-07	6.27	6.16	7.48
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	1131.2994	1.980e+17	4.941	7.481e-06	403.896	1.061e-06	6.40	6.04	5.12
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	1131.2994	1.980e+17	7.303	4.983e-06	682.038	1.061e-06	6.28	6.15	7.48
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	1131.2994	1.980e+17	6.511	2.765e-06	577.730	5.095e-07	3.37	11.46	5.12
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	1131.2994	1.980e+17	11.476	1.105e-10	5695.007	2.111e-10	2.35	16.44	7.48
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	631.2996	1.980e+17	2.466	1.520e-05	236.129	1.648e-06	10.12	3.82	2.78
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	631.2996	1.980e+17	3.784	1.315e-05	286.308	1.649e-06	9.96	3.88	4.10
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	631.2996	1.980e+17	2.540	1.754e-05	234.160	1.880e-06	10.14	3.81	2.78
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	631.2996	1.980e+17	3.858	1.516e-05	285.528	1.880e-06	9.96	3.88	4.10
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	631.2996	1.980e+17	6.308	3.657e-06	931.623	6.568e-07	2.53	15.23	2.78
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	631.2996	1.980e+17	11.324	7.789e-11	2736.034	1.153e-10	2.31	16.68	4.10
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	131.2995	1.980e+17	0.124	8.322e-05	195.794	7.461e-06	10.56	3.66	0.45
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	131.2995	1.980e+17	0.398	8.144e-05	121.570	7.461e-06	18.68	2.07	0.72
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	131.2995	1.980e+17	0.154	9.077e-05	207.529	8.858e-06	10.67	3.55	0.45
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	131.2995	1.980e+17	0.428	8.892e-05	127.505	8.857e-06	14.67	2.63	0.72
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	131.2995	1.980e+17	9.519	1.334e-06	1351.596	5.378e-07	1.80	21.47	0.45
1003.5429	4.169e+17	2.215e+17	2131.2994	1.825e+17	131.2995	1.980e+17	-0.491	1.063e-14	-45135.635	8.902e-13	0.87	44.33	0.72

Figure 37: All extracted results.