https://indico.in2p3.fr/e/simdet2020

SIMDET 2020

4th school on silicon detectors simulation

Use of Silvaco in HEP experiments

M. Bomben, APC & UPD - Paris

Université de Paris

IN2P3 Les deux infinis

Outline

- Silvaco TCAD tool
- Example: edgeless detectors
- From TCAD to Monte Carlo simulations
- Victory - From layout files to full 3D simulation
- Conclusion & Outlook

SILVACO TCAD TOOL

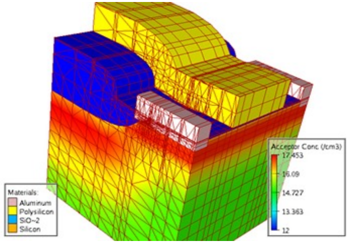
TCAD simulations

- Technology Computer Aided Design - TCAD
- Solve drift/diffusion & Poisson equations for electrons and holes:

$$J_n = qn\mu_n E + qD_n \frac{\partial n}{\partial x} \quad \frac{\partial n}{\partial t} = \frac{1}{q}\frac{\partial J_n}{\partial x} + G_n - R_n$$

$$J_p = qn\mu_p E - qD_p \frac{\partial p}{\partial x} \quad \frac{\partial p}{\partial t} = -\frac{1}{q}\frac{\partial J_p}{\partial x} + G_p - R_p$$

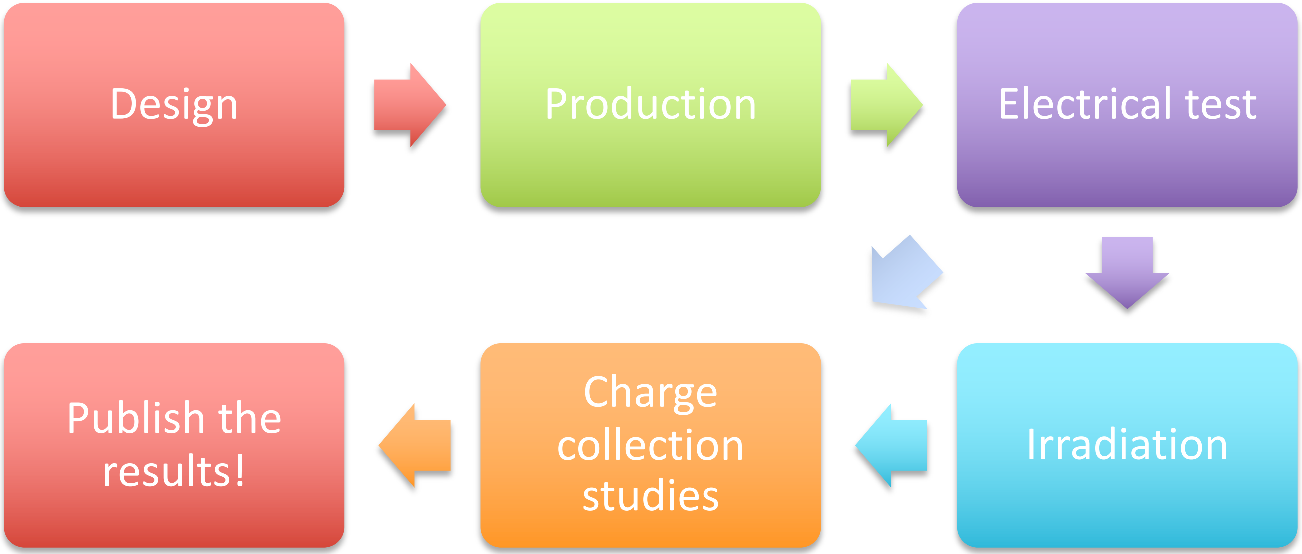
$$\frac{\partial^2 \psi}{\partial x^2} = -\frac{q}{\epsilon_{Si}\epsilon_0}(N_D + p(x) - n(x) - N_A)$$



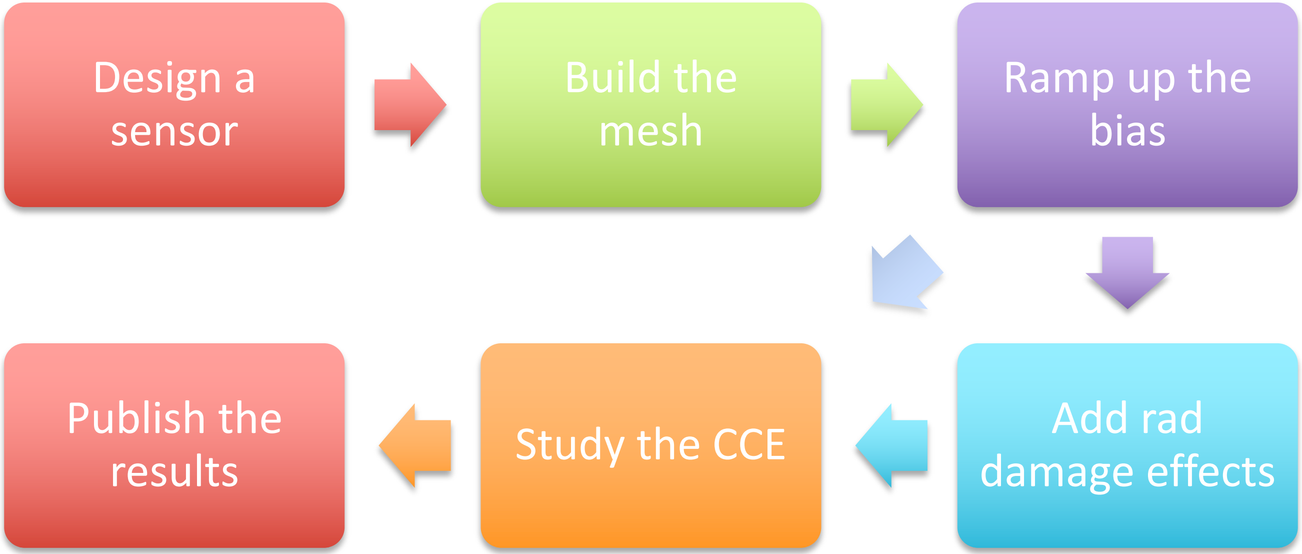
- taking into account boundary conditions
  - Electrodes' potentials, interface charges, etc
- on a grid of points

SILVACO

Normal work flow for a HEP silicon sensors



TCAD simulation work flow



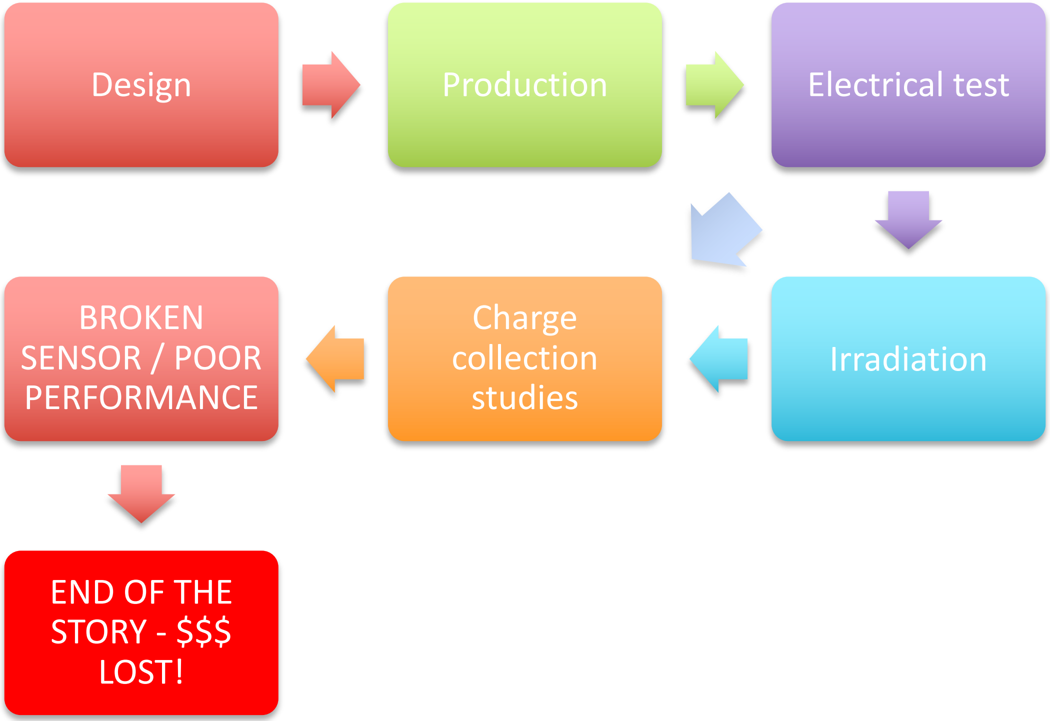
So why bother with simulations?

- You repeat all the “steps” of real sensors…

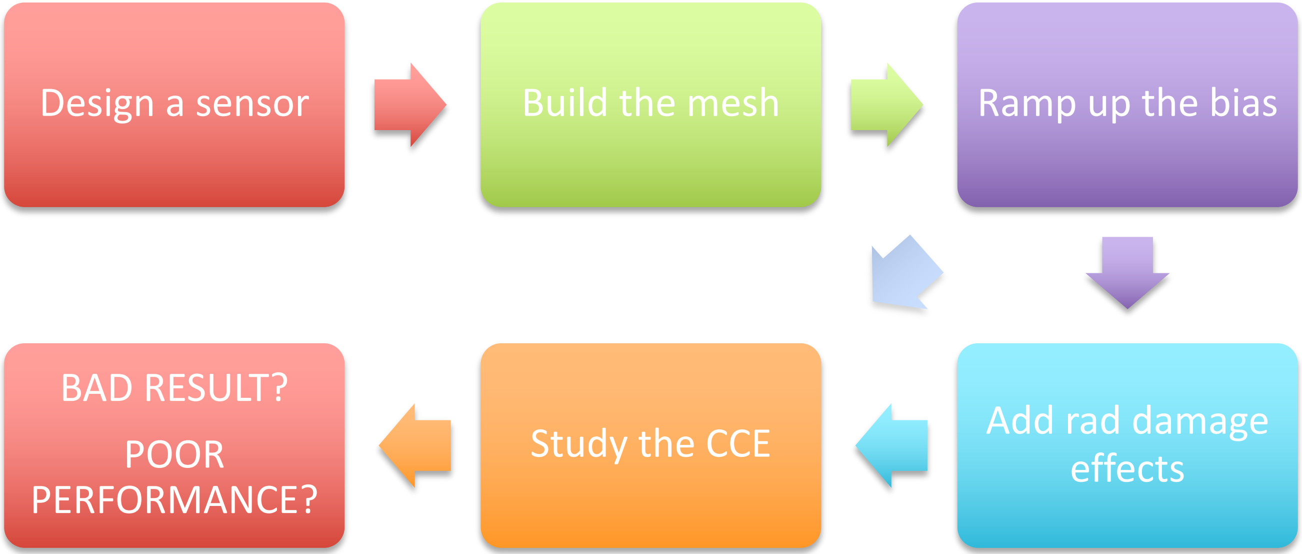
So why bother with simulations?

- You repeat all the “steps” of real sensors…
- It is not true!

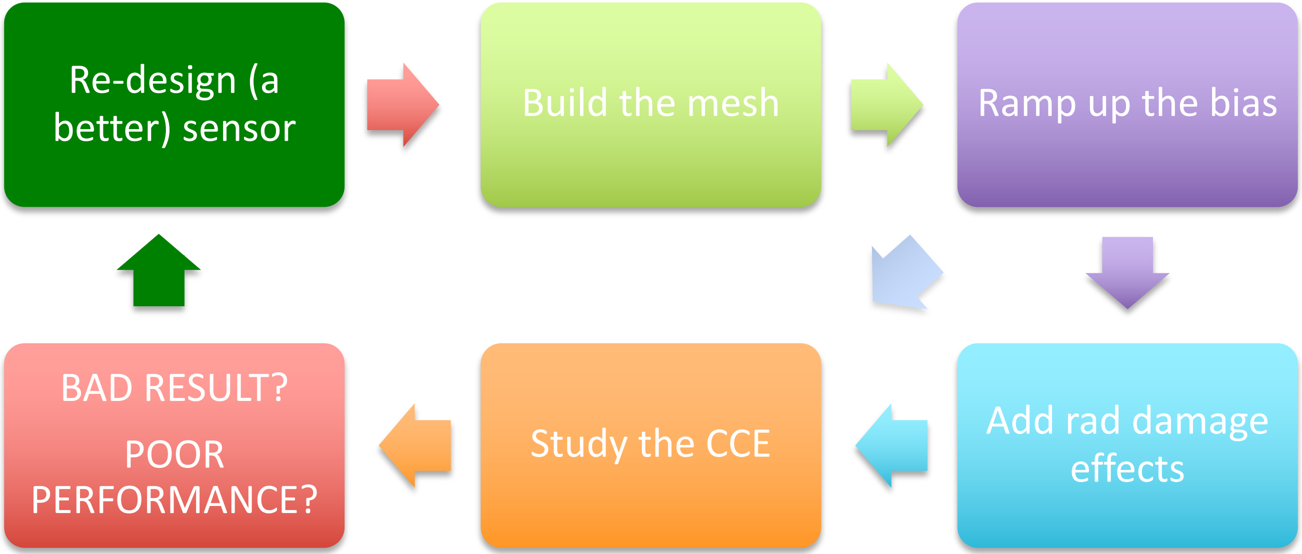
Possible work flow for real sensors



TCAD simulation work flow



TCAD simulation work flow

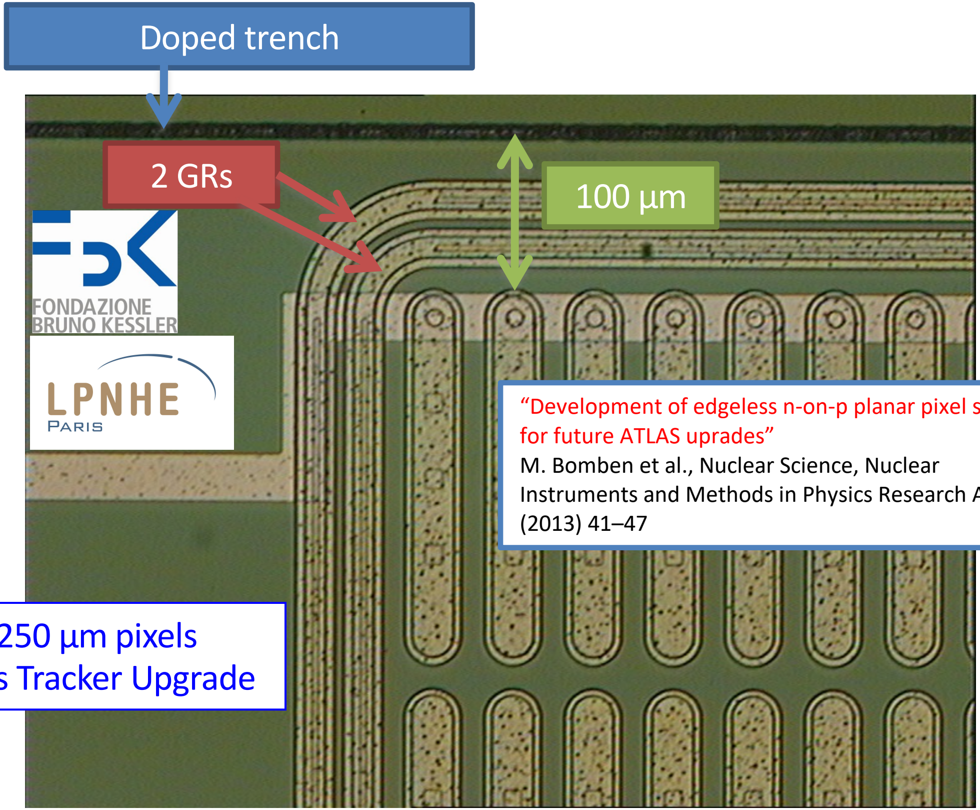


Simulations benefits

- Simulating sensors helps in saving:
  - ➢ Development time
  - ➢ Number of submissions
  - ➢ Money
- You can learn a lot in terms of:
  - ❑ Physics
    - Study quantities otherwise not accessible!
    - Examples:
      - Carrier distribution
      - Electric field distribution
      - Current densities
      - Etc.…

EXAMPLE: EDGELESS DETECTORS

Edgeless pixel detector



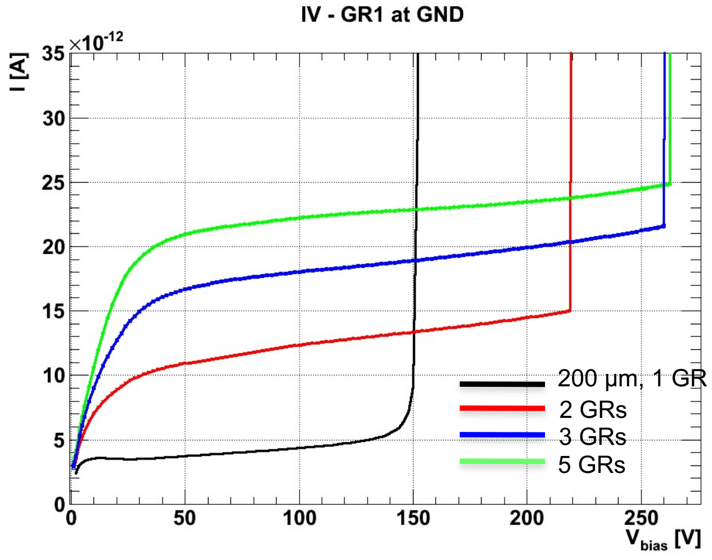
"Development of edgeless n-on-p planar pixel sensors for future ATLAS uprades"
M. Bomben et al., Nuclear Science, Nuclear Instruments and Methods in Physics Research A 712 (2013) 41–47

50 µm x 250 µm pixels
R&D for Atlas Tracker Upgrade

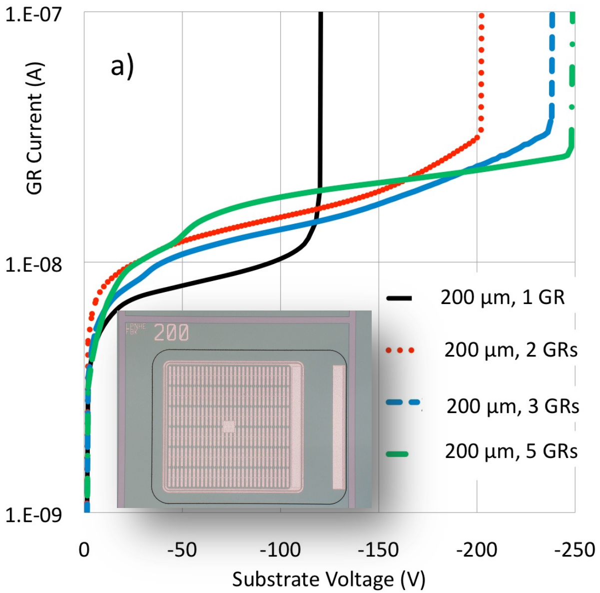
IV on test structures

Simulation drive sensor design - Focus on **breakdown (BD) voltage**

SIMULATIONS



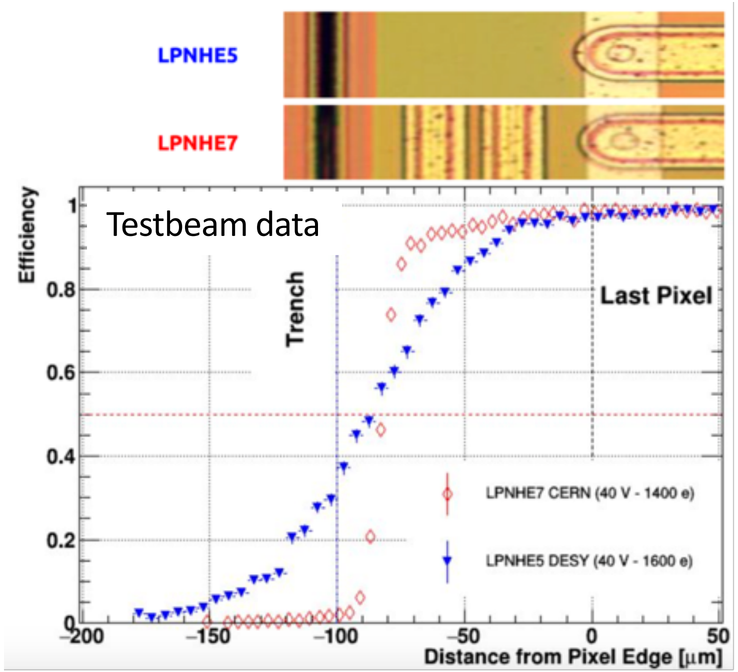
DATA

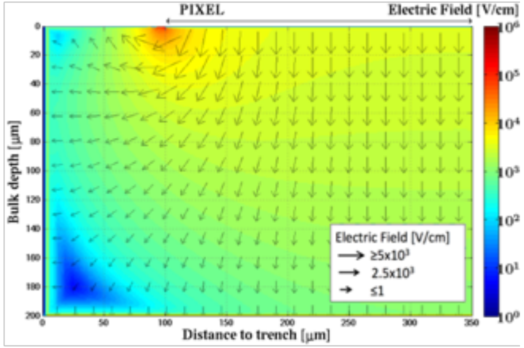


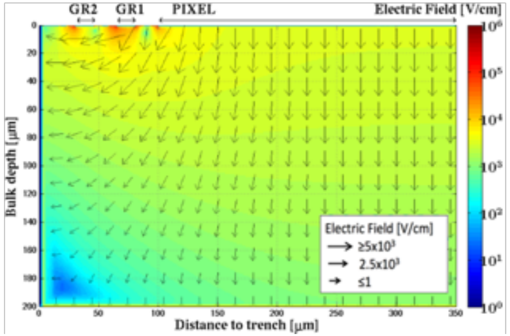
➢**BD Voltage: Agreement within 20% or better**

Hit efficiency at sensor edge

JINST 12 P05006 (2017)



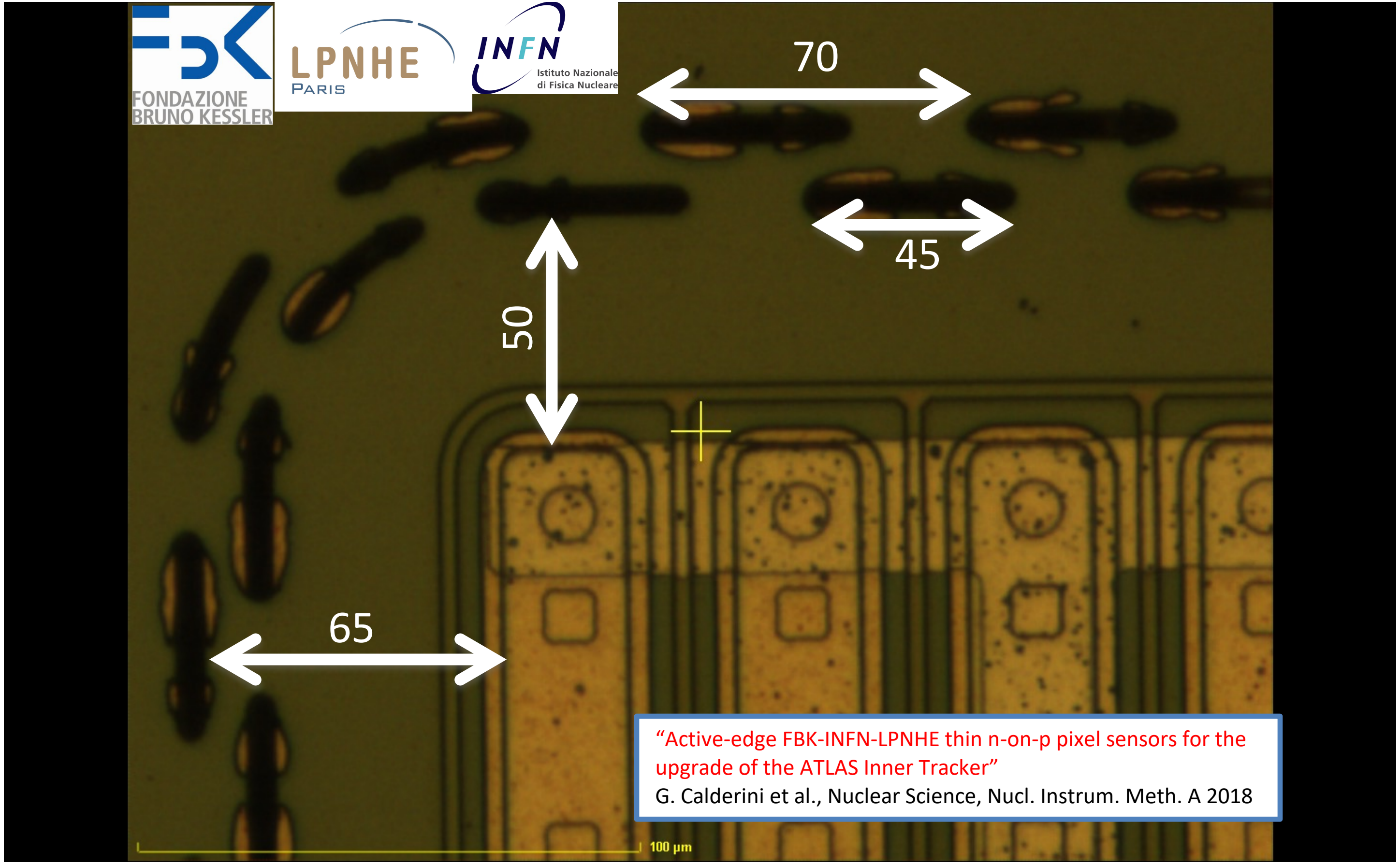




Pixel detector efficient beyond pixels area: > 80% up to 75 µm away from the last one

Reason: electric field lines closing on pixels and not on GRs!

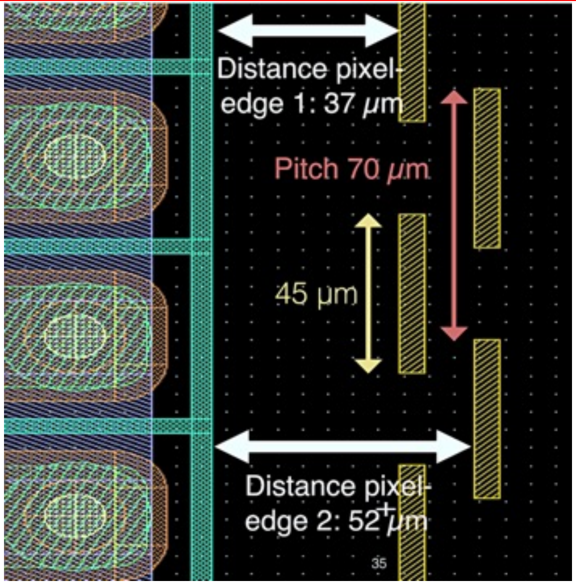
Novative edgeless production – staggered trenches

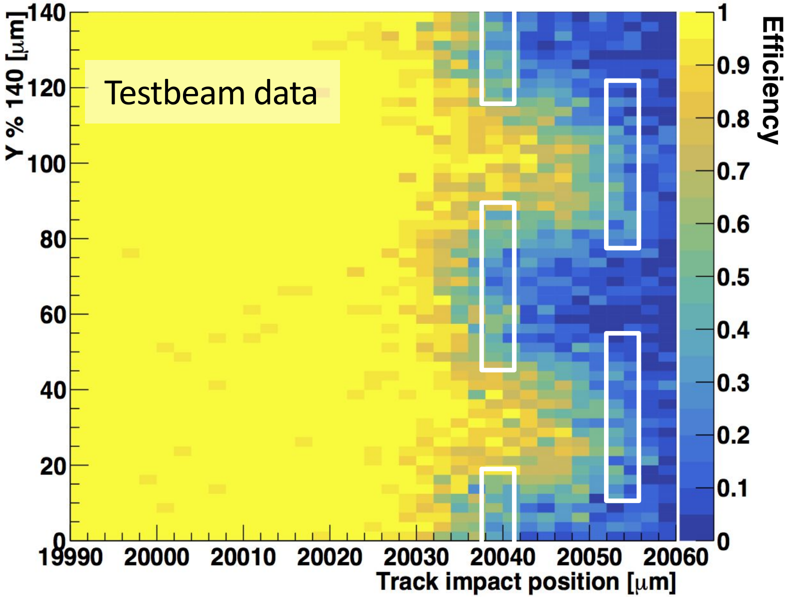


"Active-edge FBK-INFN-LPNHE thin n-on-p pixel sensors for the upgrade of the ATLAS Inner Tracker"
G. Calderini et al., Nuclear Science, Nucl. Instrum. Meth. A 2018

Hit efficiency at sensor edge

A. Ducourthial thesis
https://indico.in2p3.fr/e/18186/





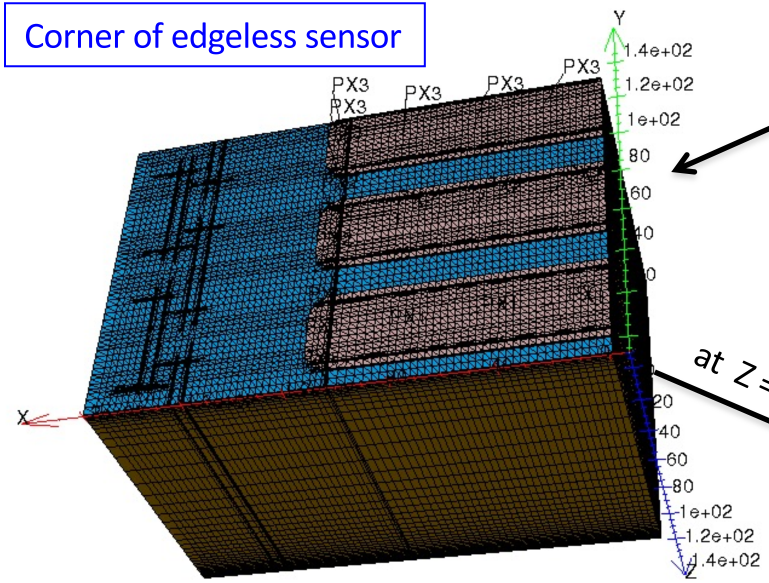
130 μm thick sensor with staggered trenches, no GRs, ~50 μm last pixel to last edge

The efficiency follows the edge pattern

The efficiency is higher than 50% up to 44 μm from the last pixel

Simulations in 3D

Corner of edgeless sensor



13439 points
26517 triangles

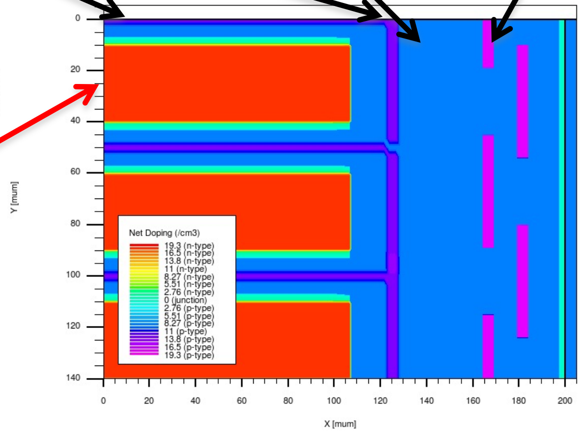
at Z = 0.2 µm

p-stop

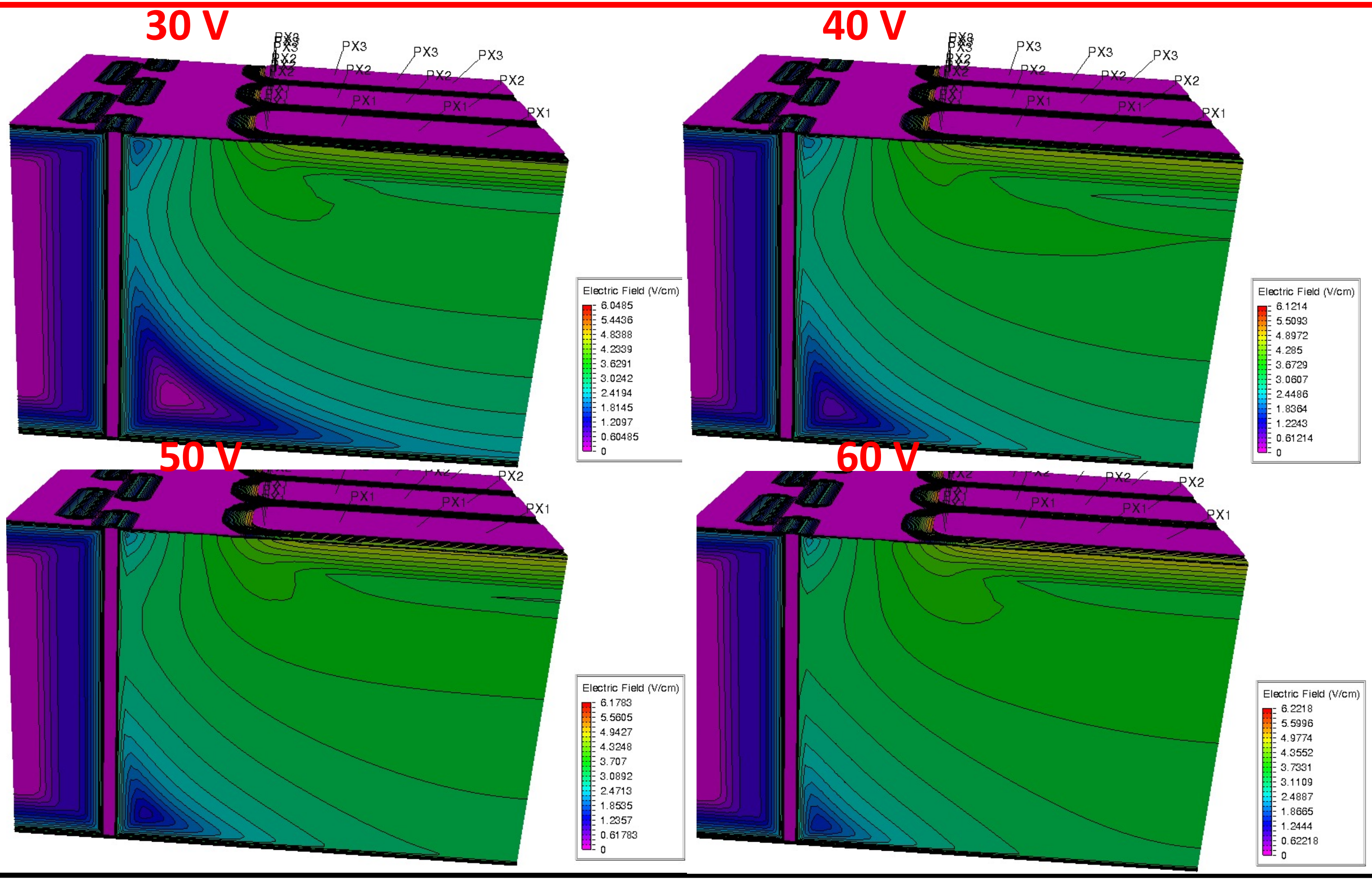
p-spray

trenches

n^+ implant



Electric field at sensor edge

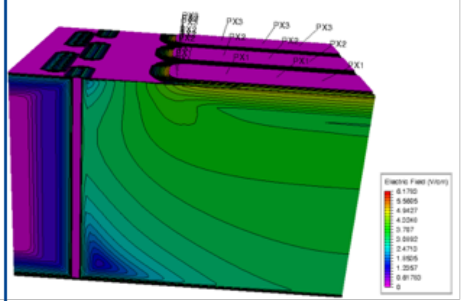


Hit efficiency at sensor edge - projections

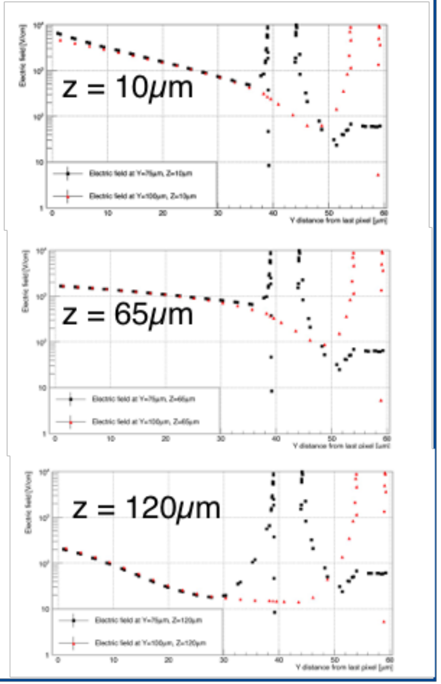
TCAD Simulations - Electric field

Simulation of the electric field for several depth z in the sensor:

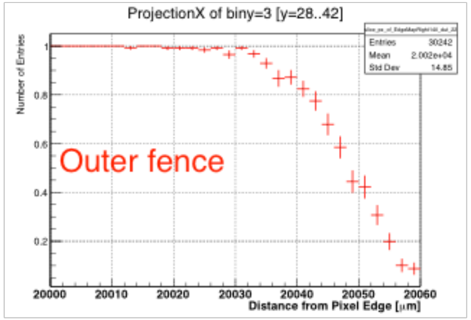
- E drops at 0 when at edge position
- low E in the bottom corner close to the edge.

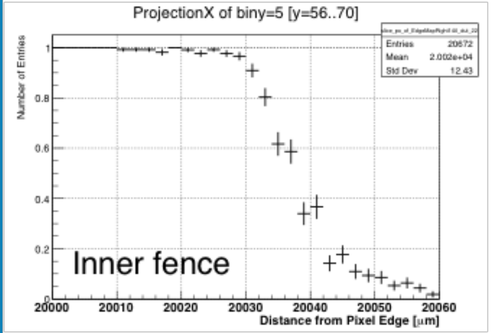


Outer fence Inner fence



Edge Efficiency





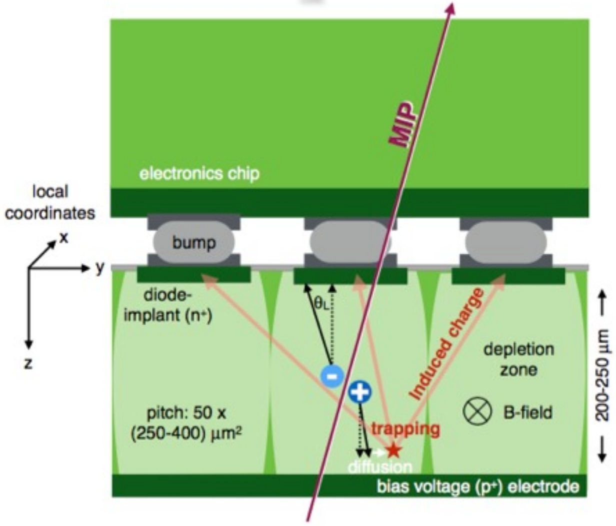
Efficiency drop matches the Electric field drop in the vicinity of the edge

FROM TCAD TO MONTE CARLO SIMULATIONS

Radiation damage effects in ATLAS MC sim.

Include all this in ATLAS MonteCarlo ✅

[*JINST* **14** P06012]



Digitization happens after simulated charge deposition and before space point reconstruction

Charge carriers will drift toward the collecting electrode due to **electric field**, which is deformed by **radiation damage**.

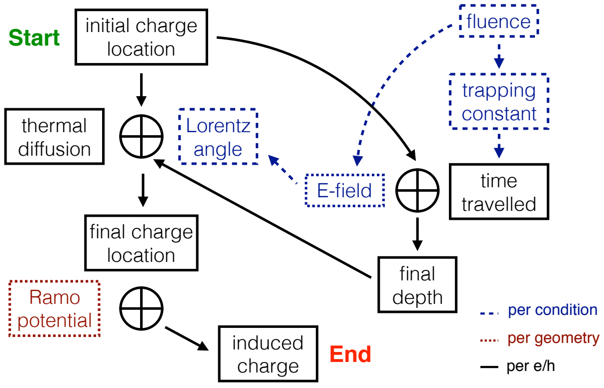
Their path will be deflected by magnetic field (**Lorentz angle**) and **diffusion**.

Due to **radiation damage** they can be **trapped** and induce/screen a fraction of their charge (**Ramo potential**).

Total induced charge is then digitized and clustered.

Implementation

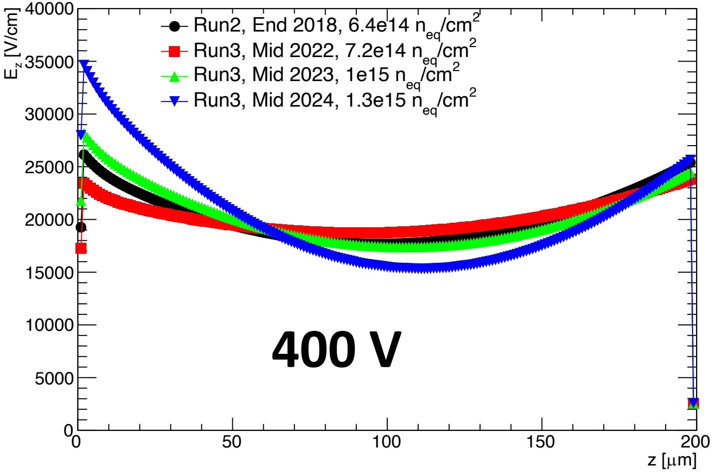
As many quantities as possible are precalculated



Ready for ATLAS MC!

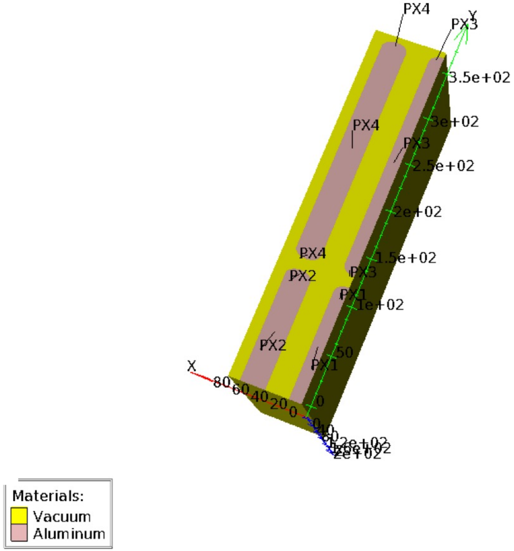
Ingredients – TCAD simulations

From fluence level the electric field is evaluated using TCAD tools

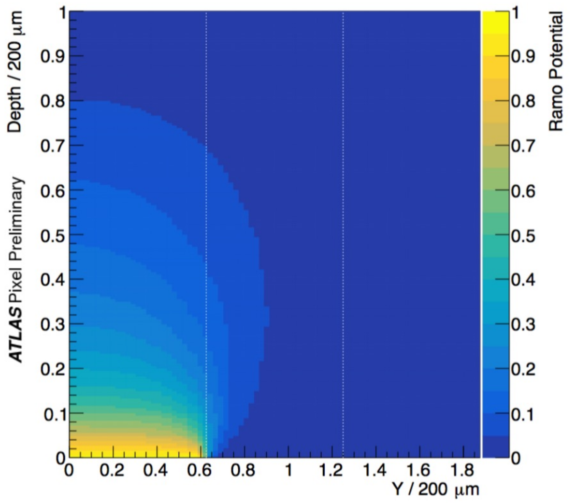


"Chiochia" model – NIM A 568 (2006)

Ramo potential from TCAD too



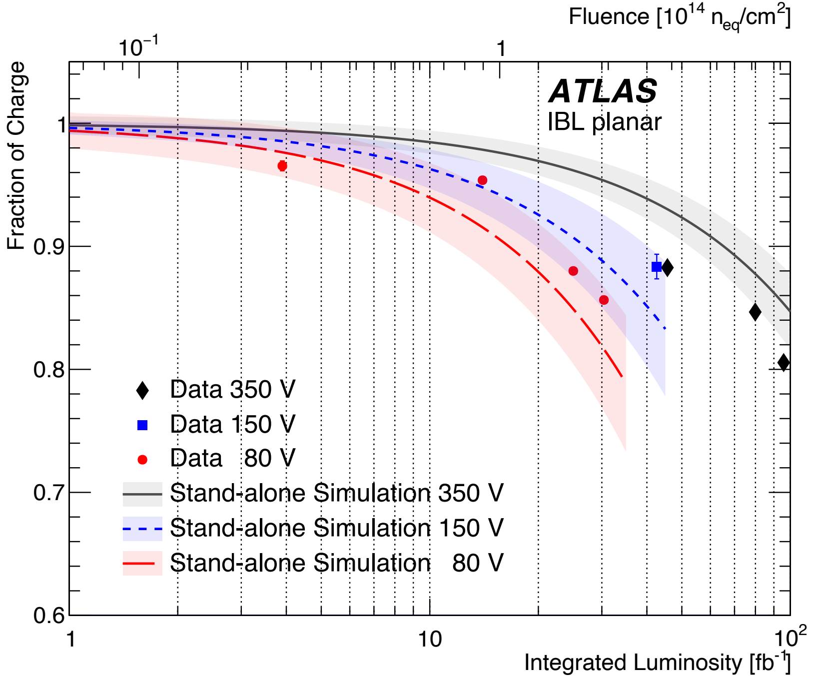
Ramo map: projection

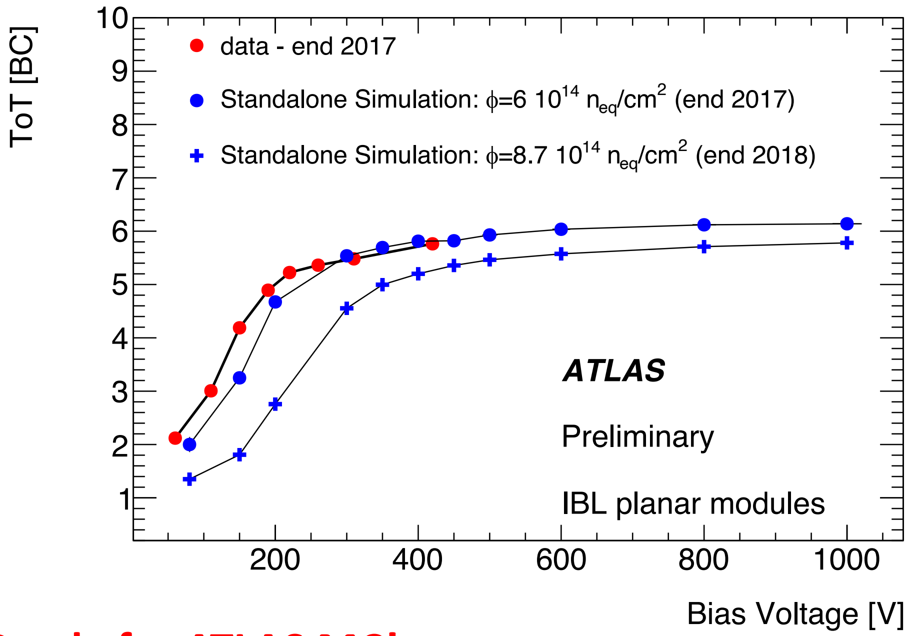


Model validation – charge collection

[*JINST* **14** P06012](#)

https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/PIX-2017-004/

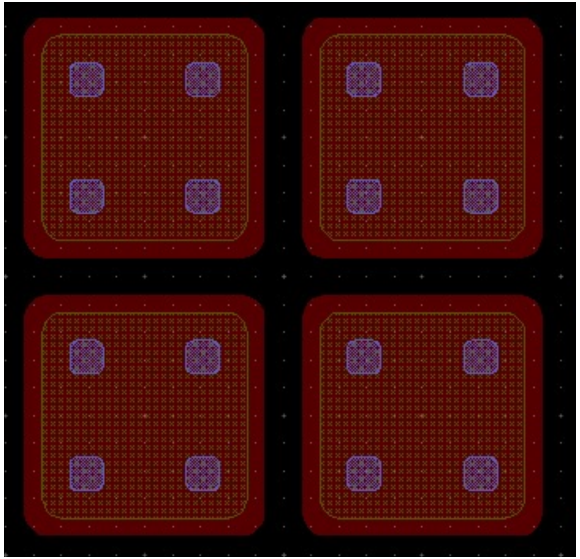


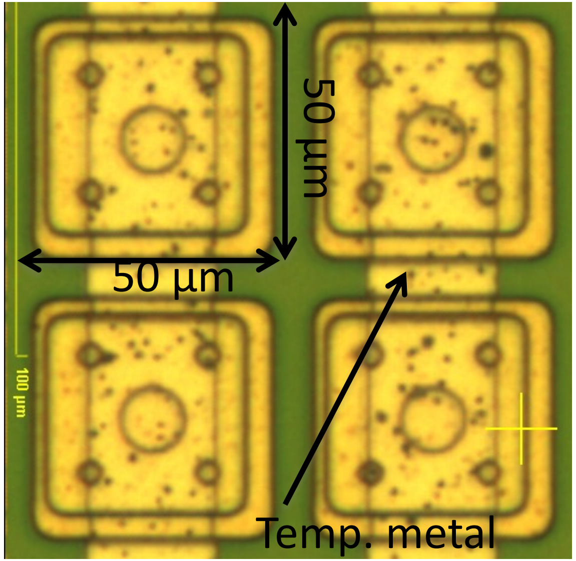


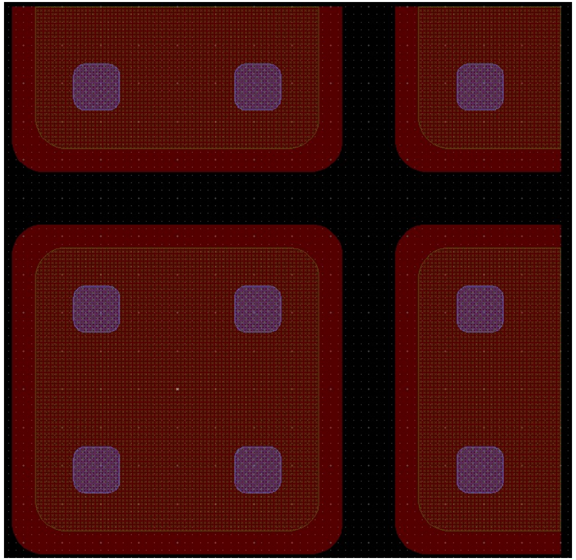
Ready for ATLAS MC!

VICTORY - FROM LAYOUT FILES TO FULL 3D SIMULATION

Fine pitch pixels







Wafer thickness: 100 µm

¼ of 3x3 pixels matrix

Victoryprocess – mesh definition

```
go victoryprocess simflags="-P 48"

#RESOLUTION/MESHDEPTH is the mesh for the regions

Init meshdepth=2 resolution=0.5,0.5 \
material=silicon c.boron=1E12 orientation=100 \
from=-25,-25 to=50,50 layout="3x3quarter.lay" \
FLOW.DIM=3D \
depth=10 gasheight=2 FLIP

deposit conformal material=oxide thick=0.5

# Line is the mesh for doping, uniform for simplicity
# define it as you would usually
Line x location=-25 spacing=3
Line x location=50 spacing=3

Line y location=-25 spacing=3
Line y location=50 spacing=3

Line z location=-2 spacing=0.4
Line z location=0.00 spacing=0.2
Line z location=3 spacing=0.2
Line z location=8 spacing=2
Line z location=8.1 spacing=0.2
Line z location=10 spacing=0.2
```

Use victoryprocess to build the structure

Define bulk material, orientation, doping, size, space on top and the layout file to be used

Deposit oxide

Add mesh lines

Simulate 10 µm thickness is enough

Victoryprocess – doping, etching, deposition

```
#backside doping
DOPING boron CONC=2e19 peak=10 gauss.sigma=0.25 lateral.sigma=0.07

# p-spray doping
DOPING boron CONC=2.5e16 peak=0 gauss.sigma=0.6 lateral.sigma=0.07

mask mask="L#1" reverse
DOPING phosphorus CONC=2e19 PEAK=0 GAUSS.SIGMA=0.4 LATERAL.SIGMA=0.07
# save name=Pbase CONFORMALSTR
#
#diffuse time=1 temperature=1000
strip

etch material=oxide thick=0.7 max mask="L#7" reverse
#save name=Etch CONFORMALSTR

deposit material=aluminum thick=0 max
#save name=via CONFORMALSTR

deposit material=aluminum thick=1 min mask="L#9"
#save name=elec CONFORMALSTR

# save simulation status for reloading into VP or VM
save name=final3D_sv

# make structure to visualize - this actually puts the conformal
# mesh on the structure so could be loaded into VD
# conformal mesh is as defined by LINE statements
save name=tmp conformalstr
```

Uniform doping

Doping through a mask

Remove mask

Etch oxide

Deposit aluminium using mask

Save for later modifications

Save for inspection/device simulation

Victoryprocess – adding electrodes

```
specifymaskpoly mask="L#91" electrode="PX1" \
    P1 = "42, 42" P2 = "42, 38" \
    P3 = "38, 38" P4 = "38, 42"
```

Add name to 4 pixels as they share a mask

```
specifymaskpoly mask="L#92" electrode="PX2" \
    P1 = "-8, 42" P2 = "-8, 38" \
    P3 = "-12, 38" P4 = "-12, 42"

specifymaskpoly mask="L#93" electrode="PX3" \
    P1 = "42, -8" P2 = "42, -12" \
    P3 = "38, -8" P4 = "38, -12"

specifymaskpoly mask="L#94" electrode="PX4" \
    P1 = "-8, -8" P2 = "-8, -12" \
    P3 = "-12, -8" P4 = "-12, -12"

electrode mask="L#91"
electrode mask="L#92"
electrode mask="L#93"
electrode mask="L#94"
```

Declare them as electrodes

```
FLIP

DEPOSIT MAX MATERIAL="Aluminium" THICKNESS=0.2

FLIP
```

Deposit aluminium on the back

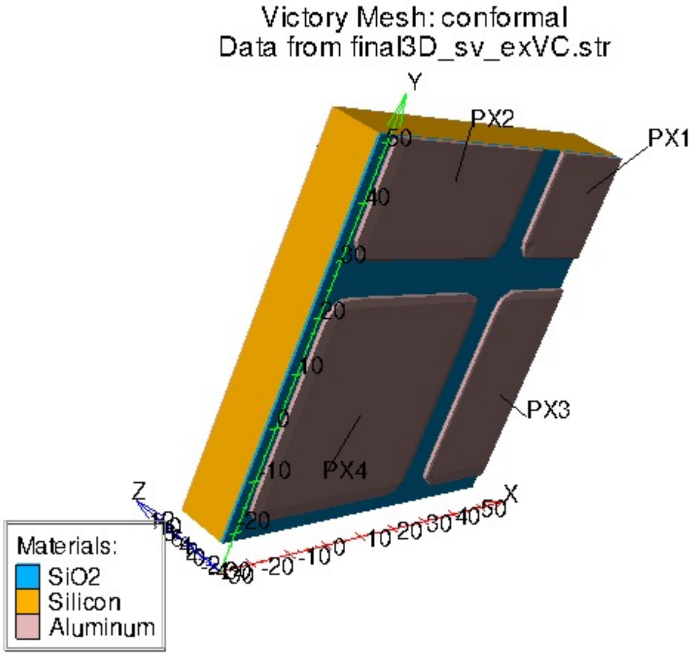
```
electrode name=HV substrate
```

Declare backside electrode

```
# save simulation status for reloading into VP or VM
save name=final3D_sv conformalstr
```

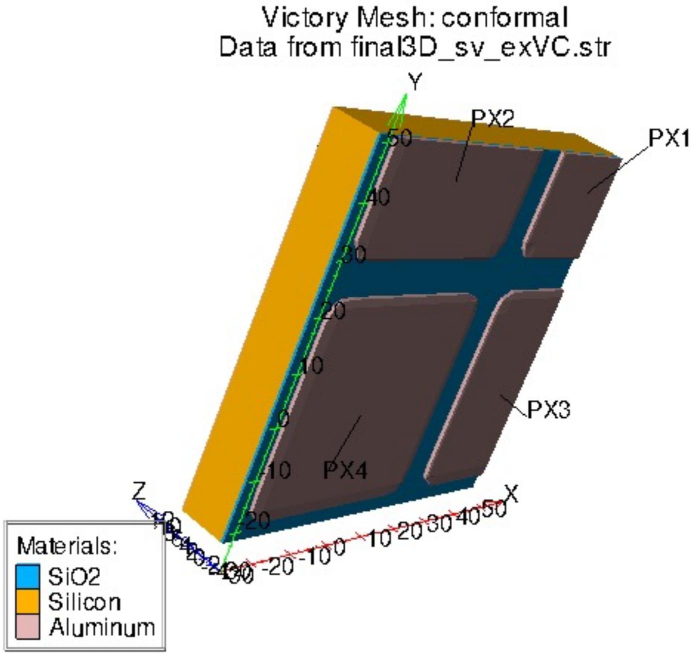
Save and ready for device simulation

Stretching structure



So far we “built” only 10 µm

Victorymesh - Stretching structure

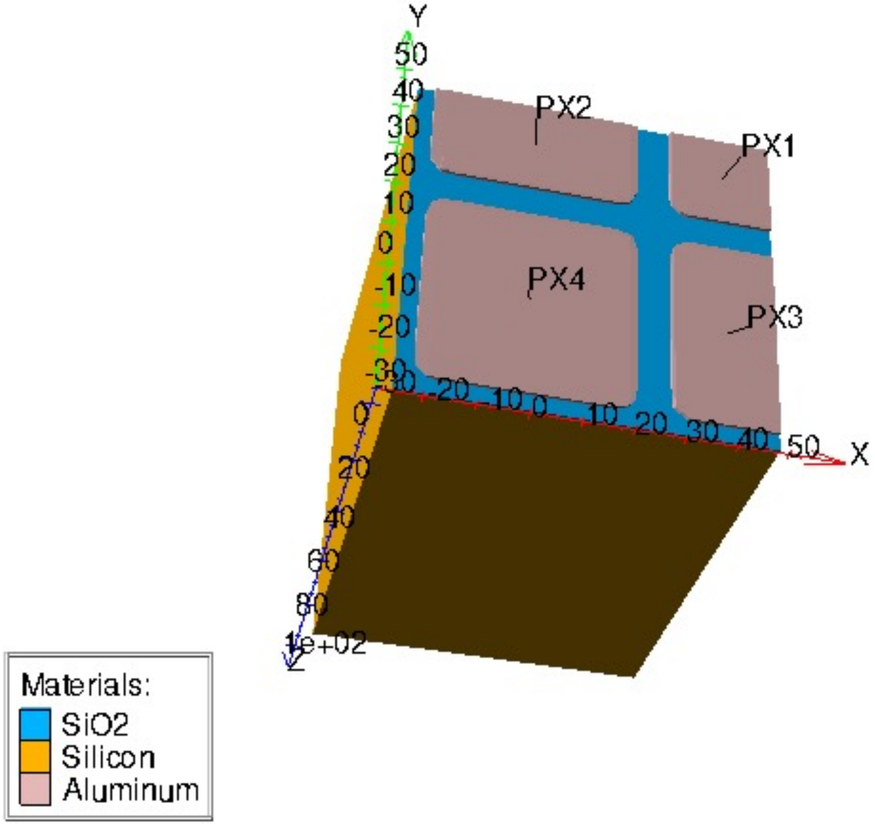


We can stretch the bulk to get the desired thickness

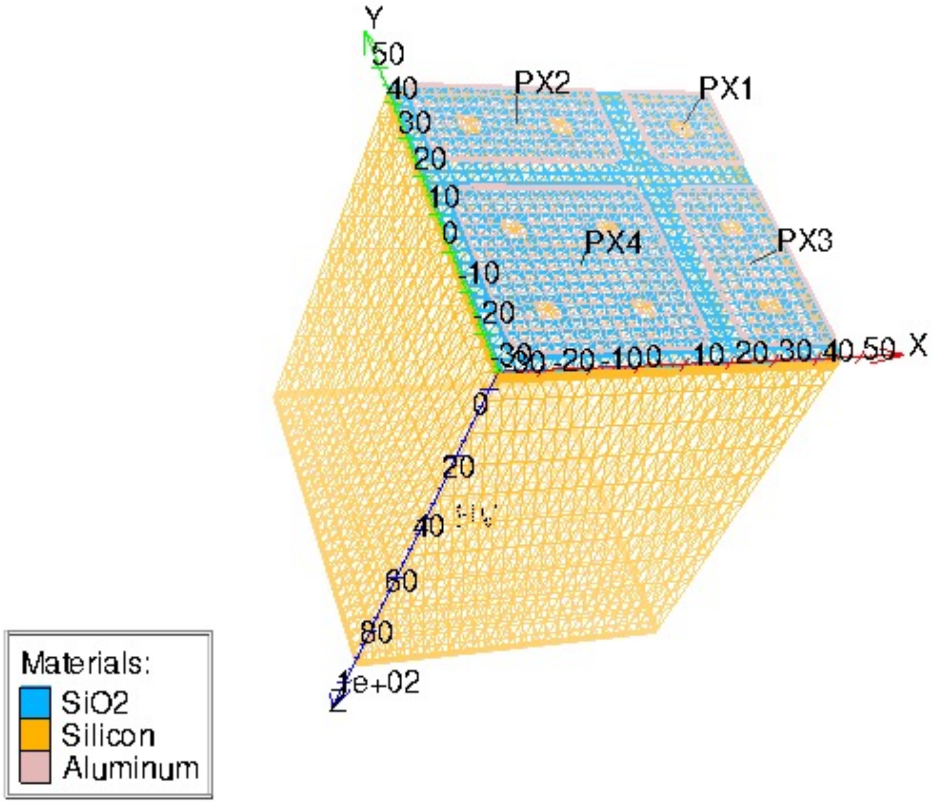
```
go victorymesh simflags="-P 32"

load in="final3D_sv"

stretch axis="z" in.intervals="3, 8" \
out.intervals="3, 98" \
axis.spacing="10"


  remesh conformal

save out="Conformal_pixel"
~
```

So far we “built” only 10 µm

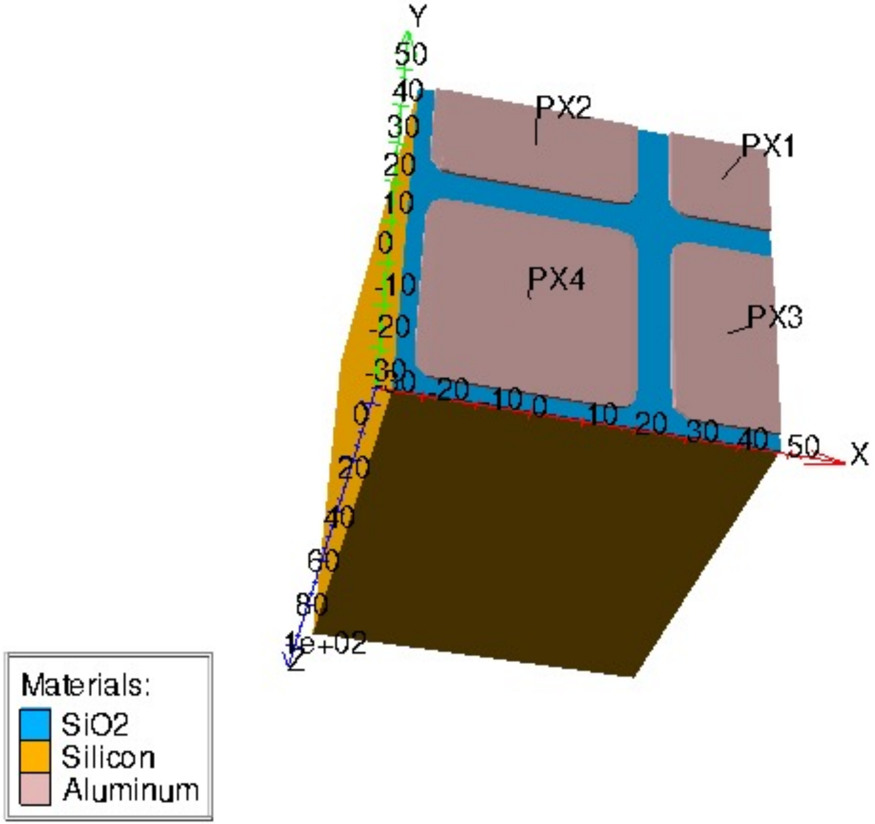
The final structure

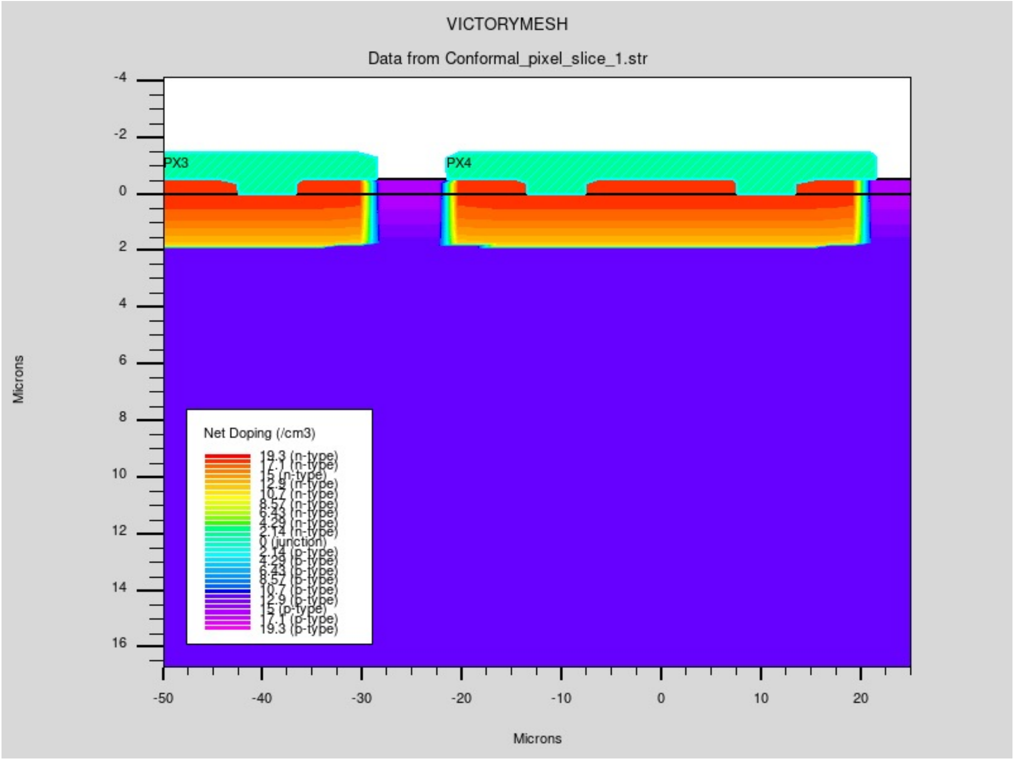


The final structure



The final structure





Victorydevice – device simulation

```
go victoryd simflags="-P 1"
assign name = solve_flags c.val="AC.ANALYSIS  Frequency=1e4"
mesh inf="../Conformal_pixel.str"
models bipolar temperature=293.15 print
interface qf=5e10
log outf="ramp.log"
solve init
solve vstep=-0.1 vfinal=-1 name=HV ${solve_flags}
save outf="ramp_1.str"
solve vstep=-1 vfinal=-5 name=HV ${solve_flags}
save outf="ramp_5.str"
solve vstep=-1 vfinal=-10 name=HV ${solve_flags}
save outf="ramp_10.str"
solve vstep=-5 vfinal=-50 name=HV ${solve_flags}
save outf="ramp_50.str"
solve vstep=-10 vfinal=-100 name=HV ${solve_flags}
save outf="ramp_100.str"
quit
```

Declare variables

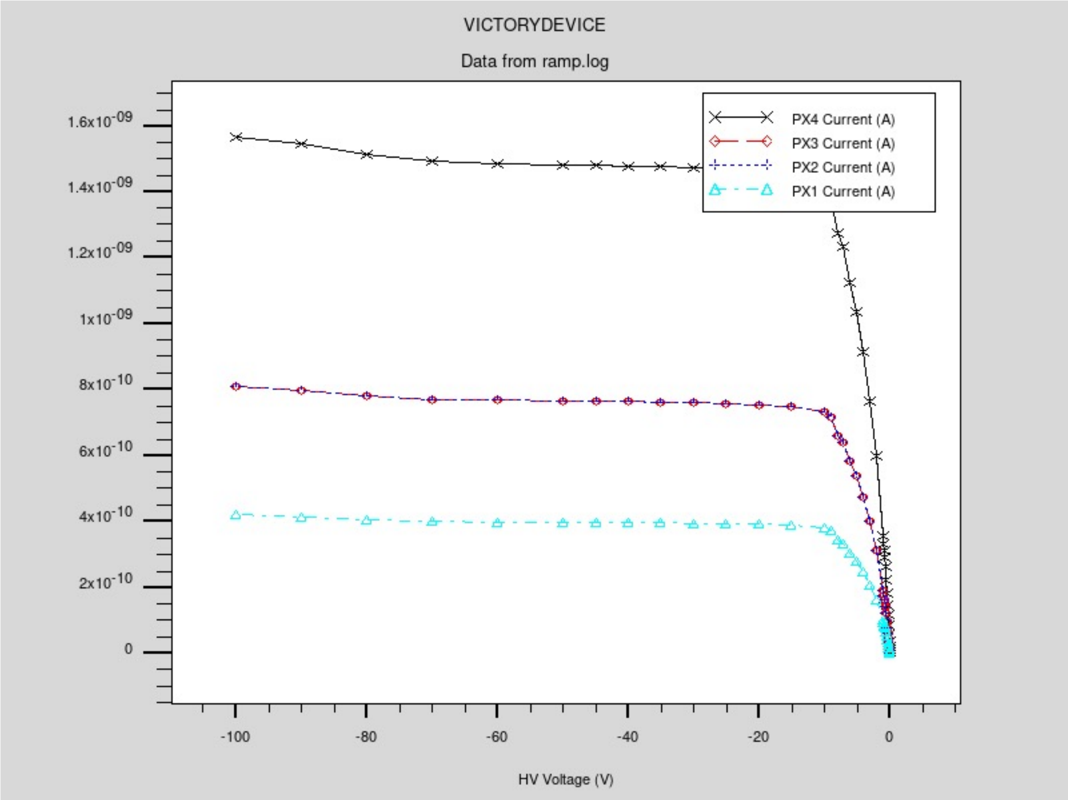
Load structure, define physics models and temperature

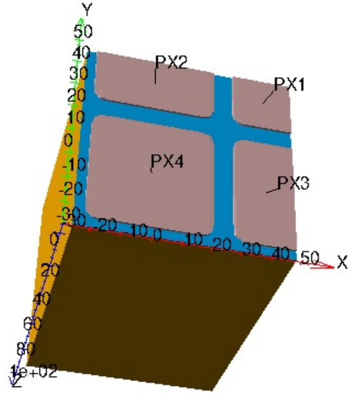
Add interface charge

Start simulating for V=0 on all electrodes

Ramp voltage, perform AC simulation and save solutions each time you want
(Later you can restart simulating from such solution files)

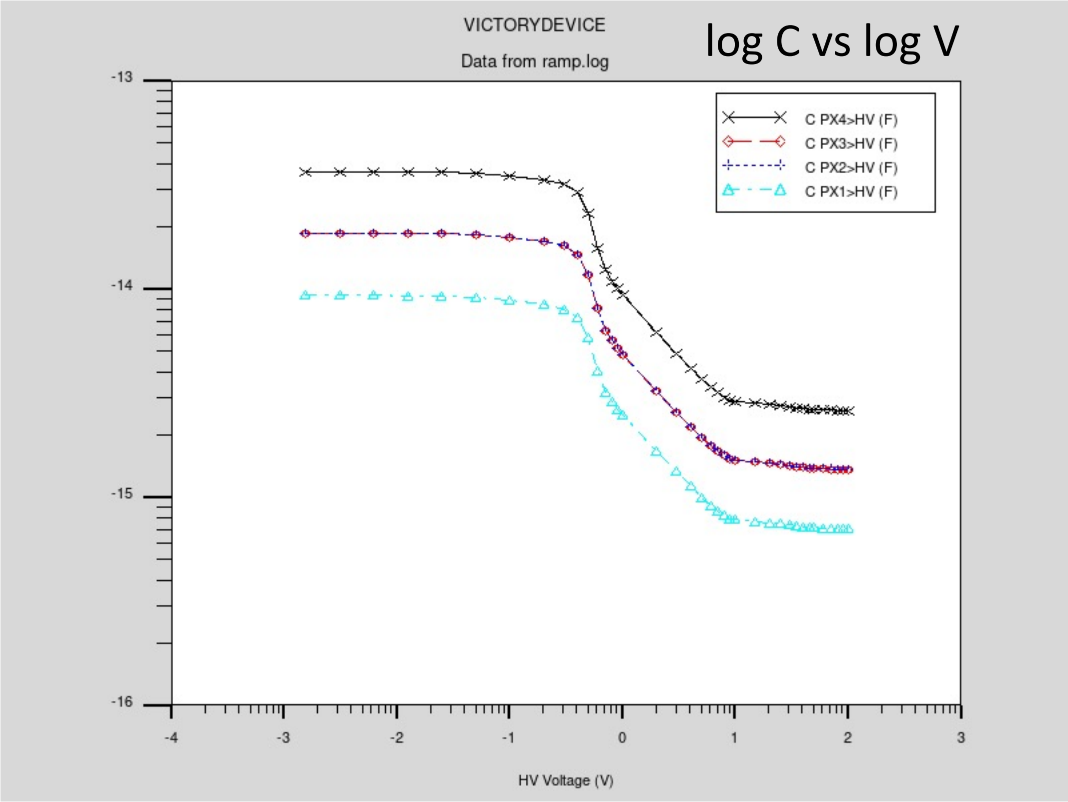
Tonyplot(3D) – Leakage currents

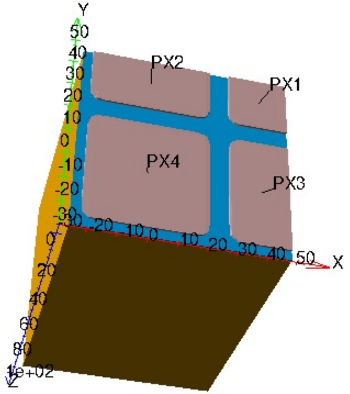




Current level matches theoretical predictions

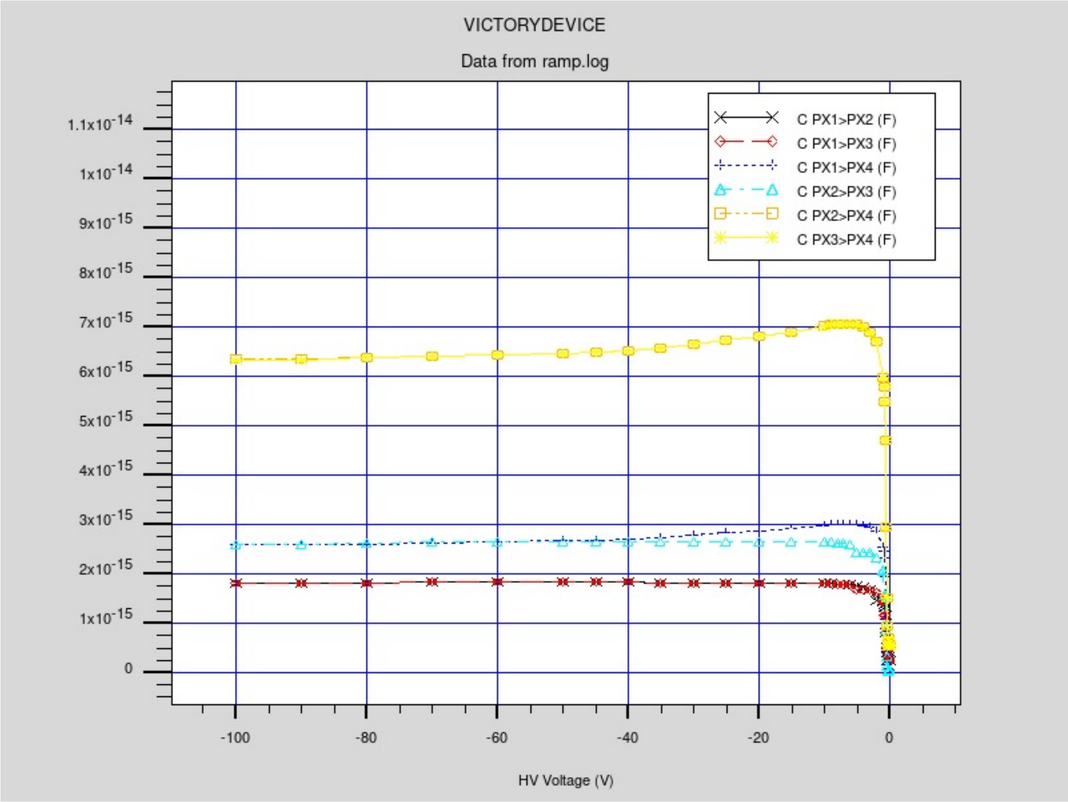
Tonyplot(3D) – depletion capacitance

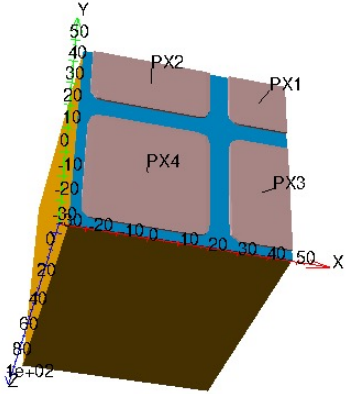




Depletion voltage and capacitance at depletion in agreement with expectations

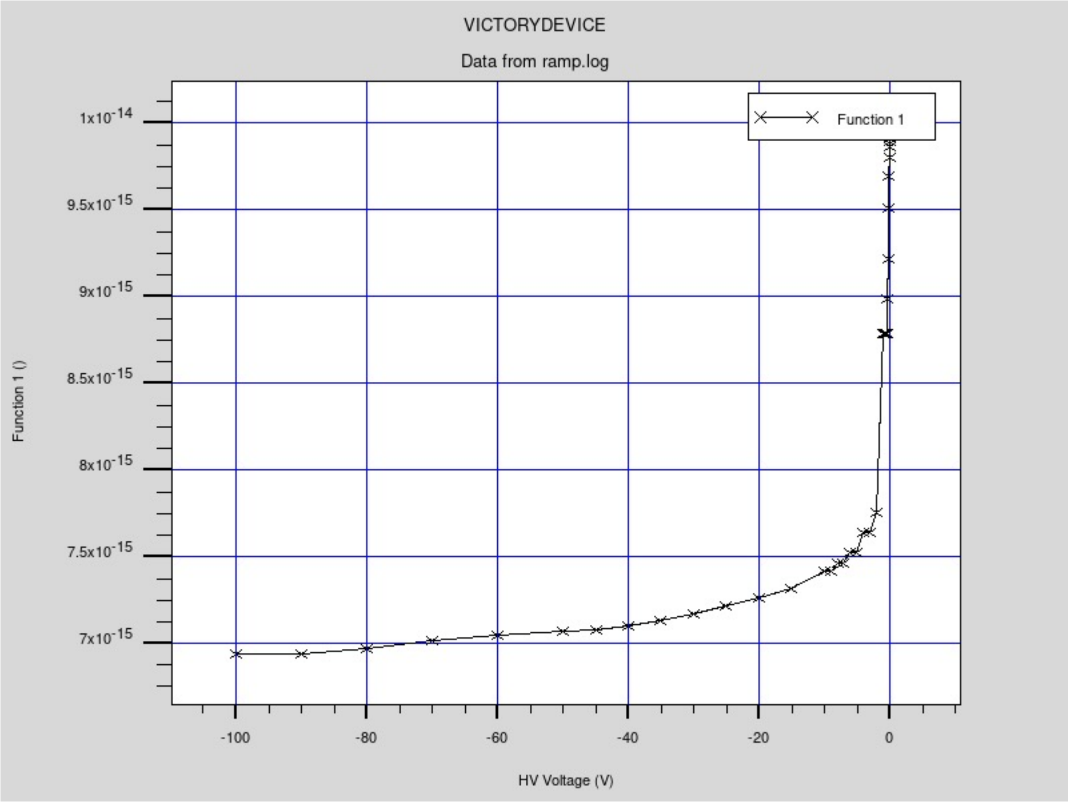
Tonyplot(3D) – Interpixel capacitance

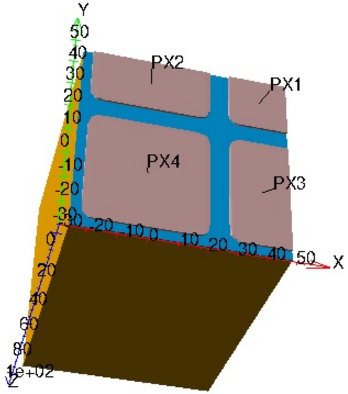




Nice scaling with shared perimeter

Tonyplot(3D) – Total input capacitance





Valuable information for many applications!

Victorydevice – Transient signal simulation

Define few variables

Read structure file

Declare physics models and everything as in the ramp simulation

Solve again for the selected bias point

```
## bias point
set bias = 50
### entry_point
set x_entry_point = 0
set y_entry_point = 0
### entry_point
set x_exit_point = 0
set y_exit_point = 0

set file_name="pixel_3D"

set log_file_name="pixel_3D"


go victoryd simflags="-P 8"
mesh inf="../../Conformal_pixel.str"

models bipolar temperature=293.15 print

interface qf=5e10

log outf="${log_file_name}.log"
load infile="../ramp_${bias}.str" master
solve prev

#method halfimplicit

# Specify the charge track: normal incidence through the drain
singleeventupset entry="${x_entry_point},${y_entry_point},0" exit="${x_exit_point},${y_exit_point},100" pcunits b.density=2.18e-5 \
                 radialgauss radius=5  t0=2.e-11 tc=0
# Log file for transient
assign name = log_file_name c.val="$'log_file_name'-SEU"
log outf="$'log_file_name'.log"
assign name = file_name c.val="$'file_name'-SEU"
```

Victorydevice – Transient signal simulation

```
## bias point
set bias = 50
### entry_point
set x_entry_point = 0
set y_entry_point = 0
### entry_point
set x_exit_point = 0
set y_exit_point = 0

set file_name="pixel_3D"

set log_file_name="pixel_3D"


go victoryd simflags="-P 8"
mesh inf="../../Conformal_pixel.str"

models bipolar temperature=293.15 print

interface qf=5e10

log outf="${log_file_name}.log"
load infile="../ramp_${bias}.str" master
solve prev

#method halfimplicit

# Specify the charge track: normal incidence through the drain
singleeventupset entry="${x_entry_point},${y_entry_point},0" exit="${x_exit_point},${y_exit_point},100" pcunits b.density=2.18e-5 \
                 radialgauss radius=5  t0=2.e-11 tc=0
# Log file for transient
assign name = log_file_name c.val="$'log_file_name'-SEU"
log outf="$'log_file_name'.log"
assign name = file_name c.val="$'file_name'-SEU"
```

Declare entry/exit point of charge deposition and the charge density

Save transient signals

Victorydevice – Transient signal simulation

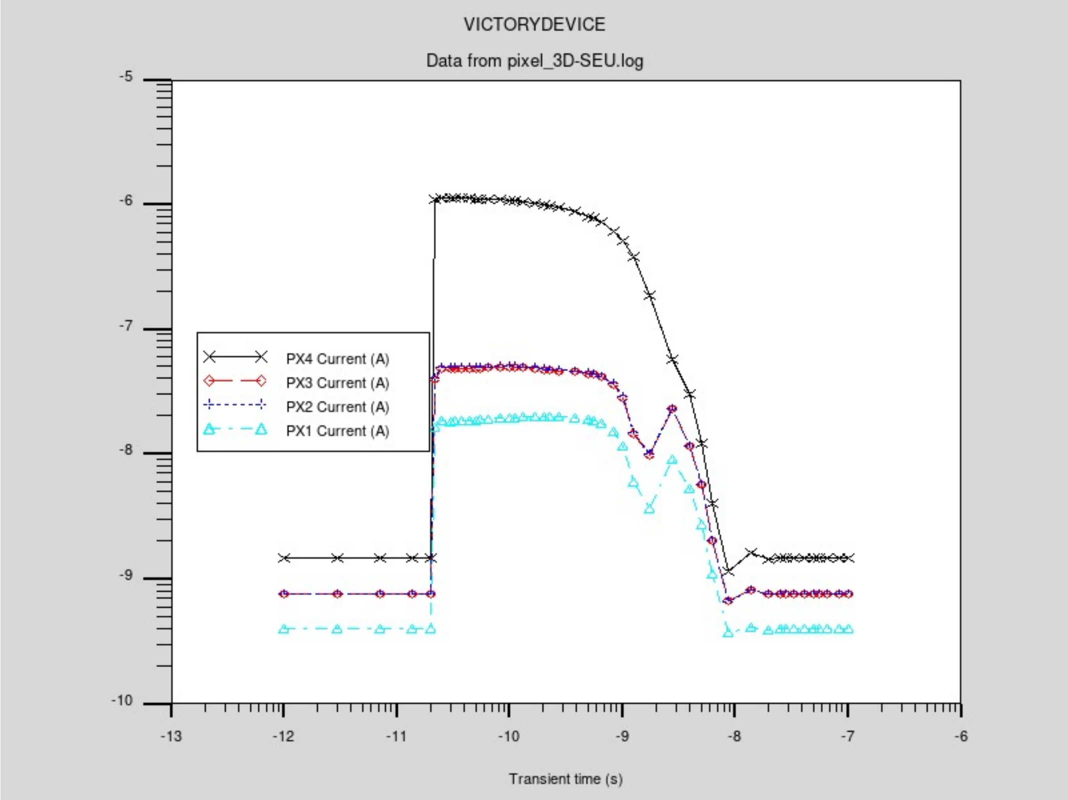
```
## seu peak
#method constant.timestep
solve tfinal=2.0e-11 tstep=1.0e-12
save outf="$'file_name'-before-seu.str"
#
## Response to particle strike
#method lte.timestep tstep.incr=1.25 dt.max=2.5e-11
solve tfinal=3e-11 tstep=1.5e-12 prev
save outf="$'file_name'-during-seu.str"
#
## after 50 ps
#method lte.timestep tstep.incr=1.25 dt.max=2.5e-11
solve tfinal=5e-11 tstep=2.5e-12 prev
save outf="$'file_name'-after-50ps.str"
#
#
## after 100 ps
#method lte.timestep tstep.incr=1.25 dt.max=2.5e-11
solve tfinal=1e-10 tstep=5.0e-12 prev
save outf="$'file_name'-after-100ps.str"
#
#
## after 200 ps
#method lte.timestep tstep.incr=1.25 dt.max=2.5e-11
solve tfinal=2.0e-10 tstep=1.0e-11 prev
save outf="$'file_name'-after-200ps.str"
#
## after 500 ps
solve tfinal=5.0e-10 tstep=2.5e-11 prev
save outf="$'file_name'-after-500ps.str"
#
## after 1 ns
solve tfinal=1e-9 tstep=5.0e-11 prev
save outf="$'file_name'-after-1ns.str"
#
```

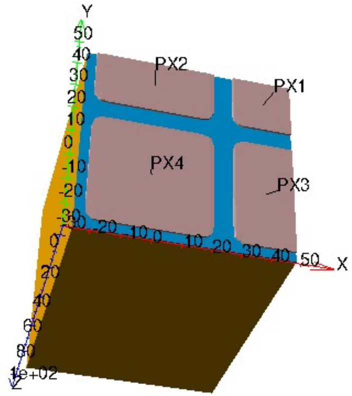
Solve as a function of time by defining the final and incremental time

Save the simulated structure

Repeat as many times as needed, to capture the evolution of many observables (concentrations, current densities, generation rates, etcetera)

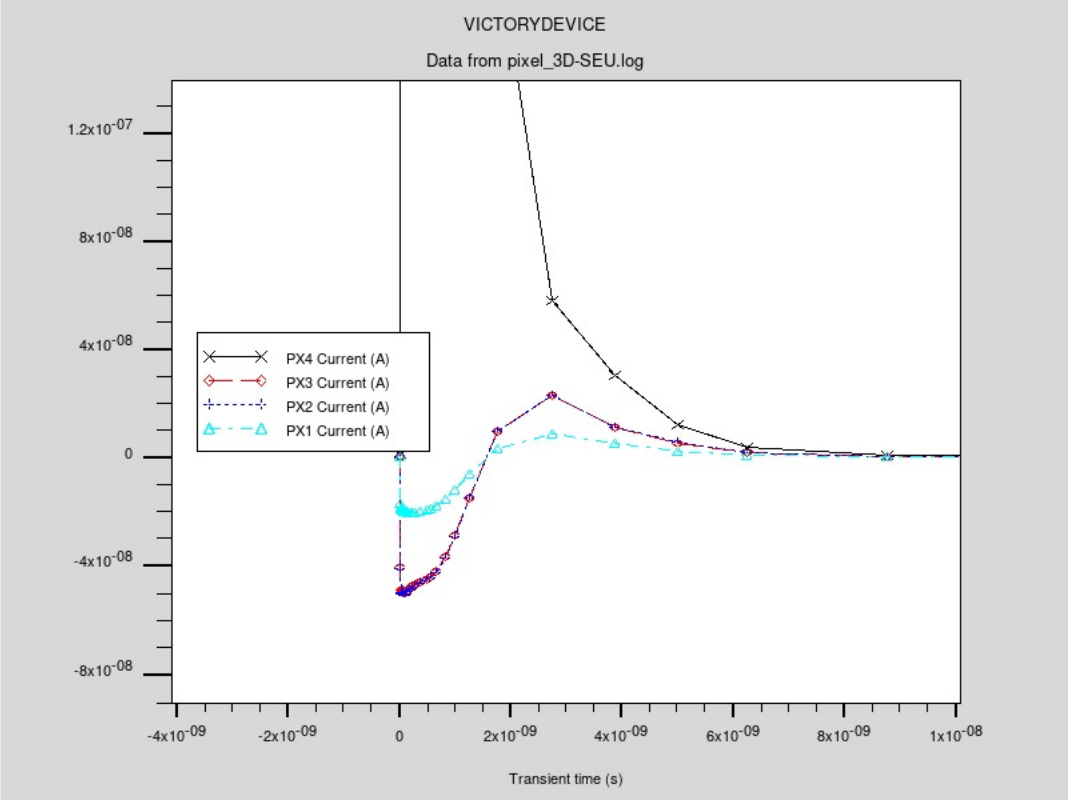
Tonyplot – Transient currents

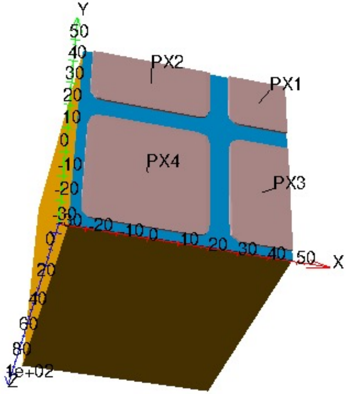




Particle striking in the middle of PX4

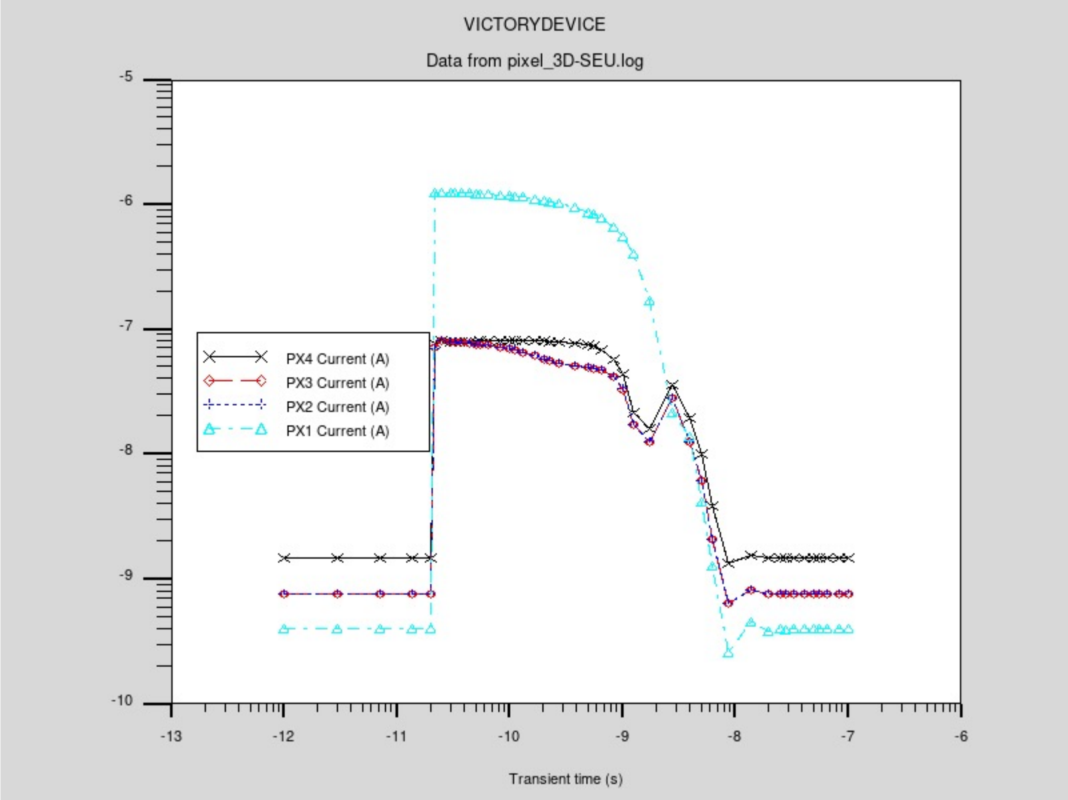
Tonyplot – Transient currents

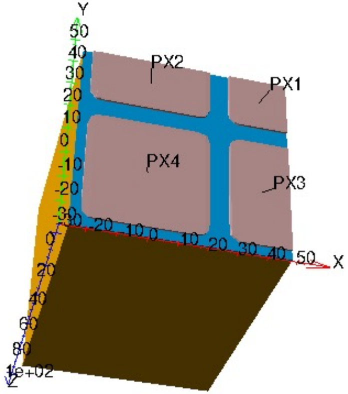




Opposite signal induced on neighbours

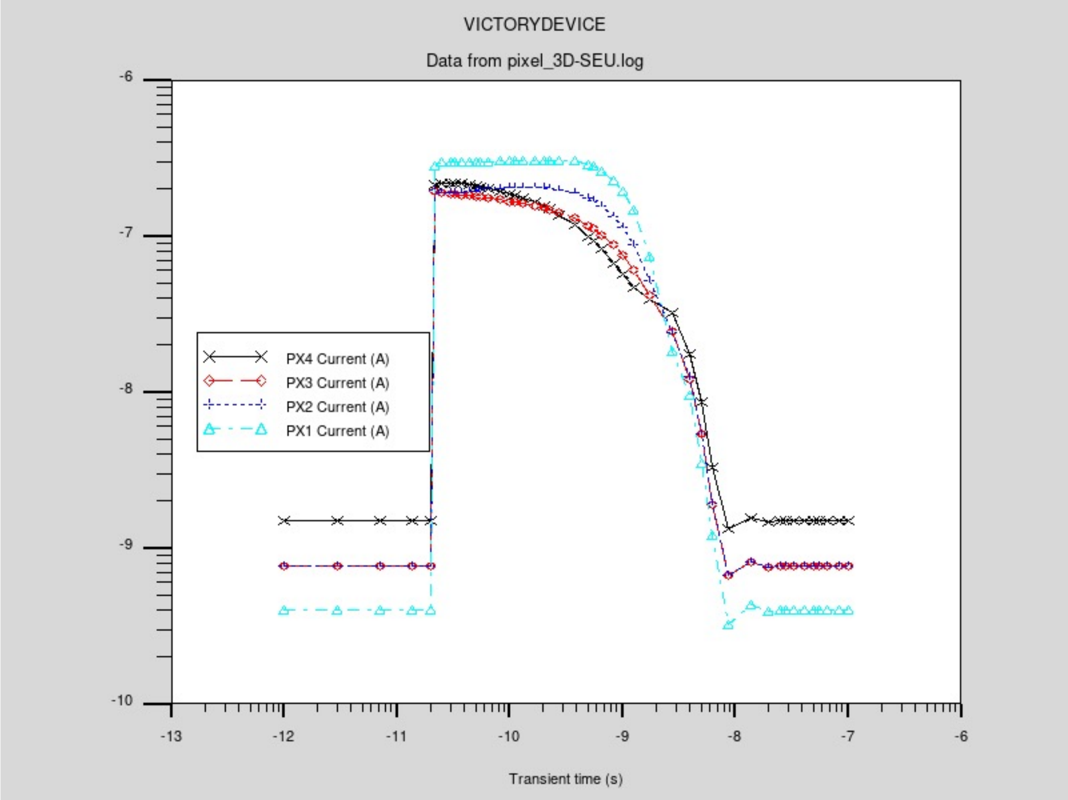
Tonyplot – Transient currents

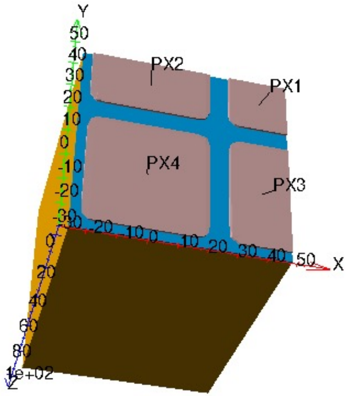




Particle striking in the middle of PX1

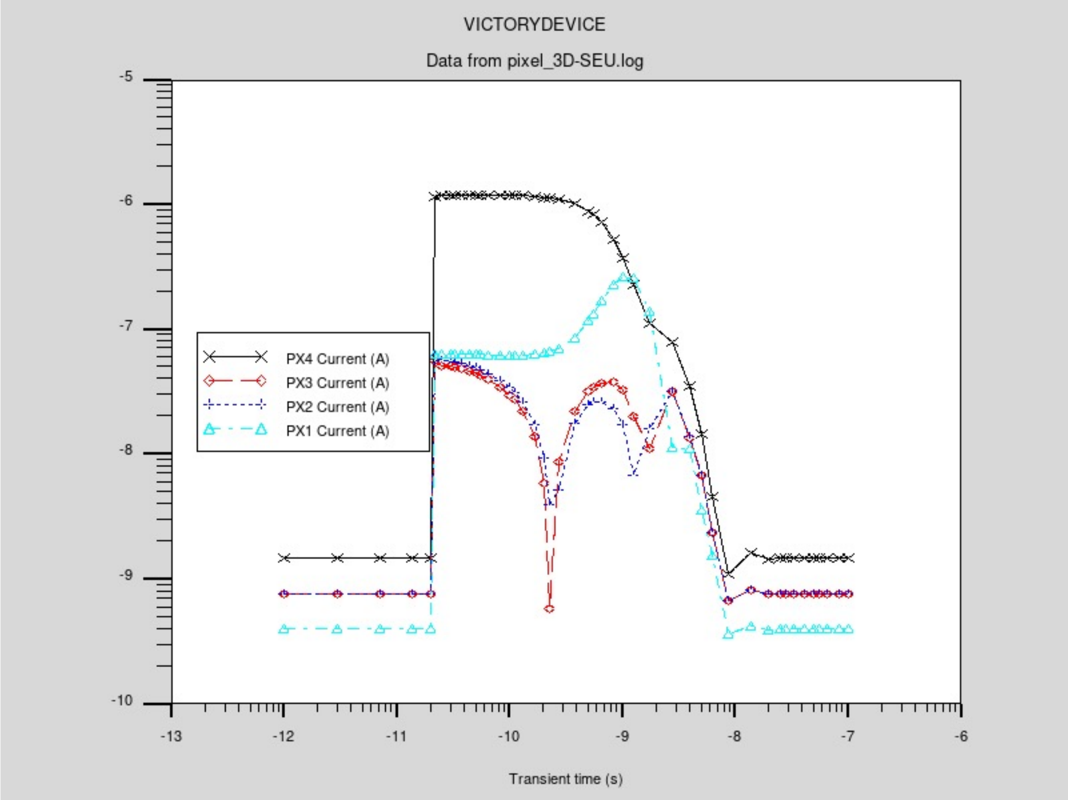
Tonyplot – Transient currents

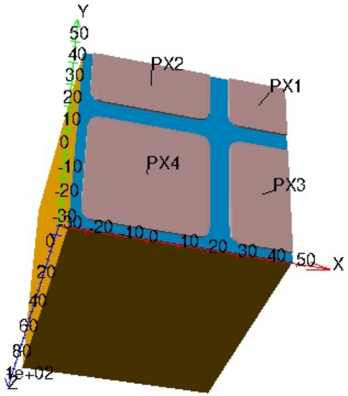




Particle striking in between pixels

Tonyplot – Transient currents



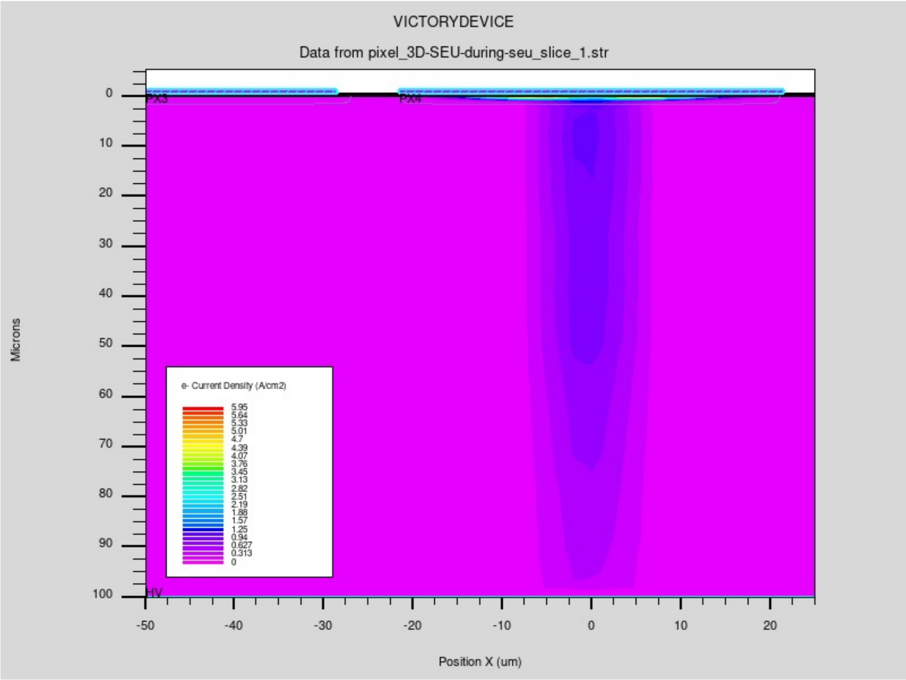


Particle striking diagonally

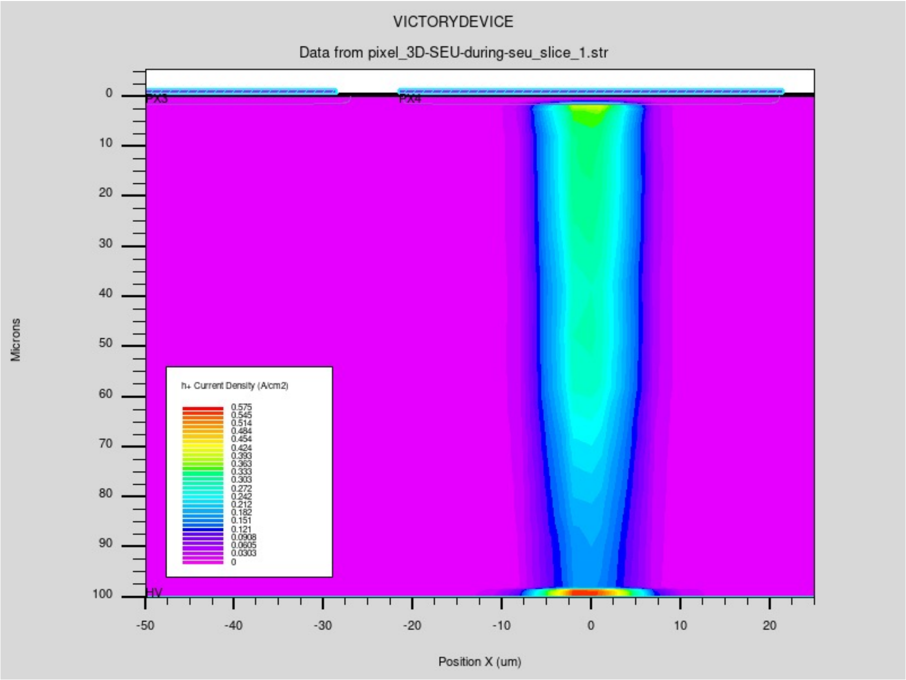
Tonyplot – Current densities

1.5 ps after particle strike

e-



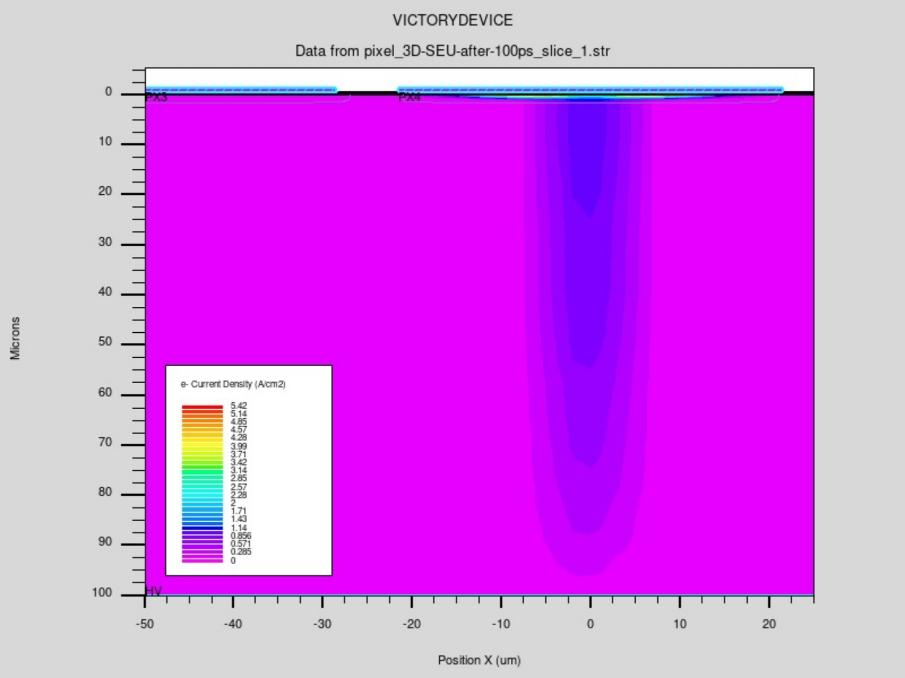
h+



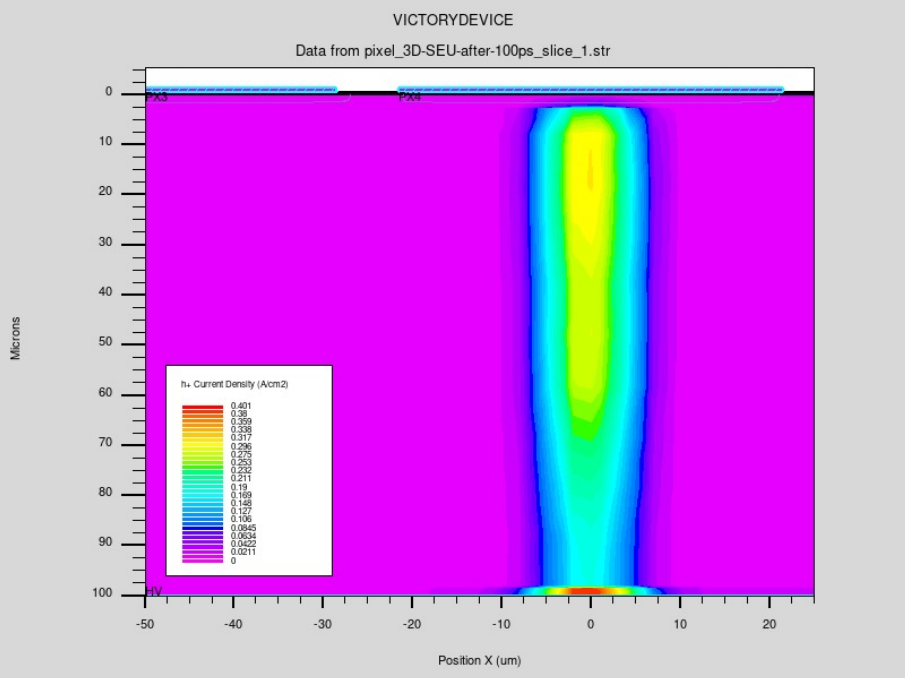
Tonyplot – Current densities

100 ps after particle strike

e-

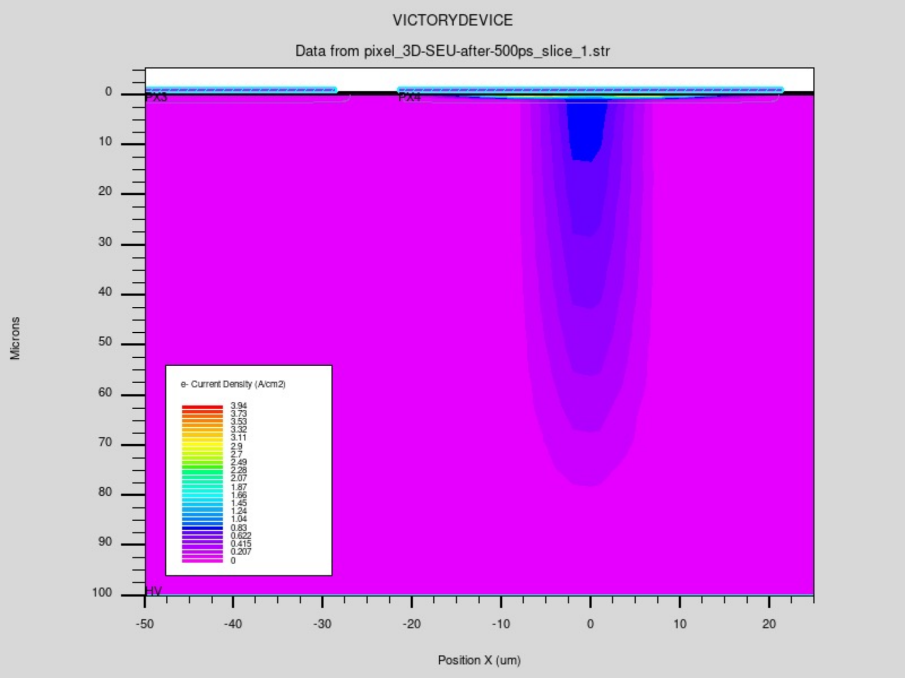


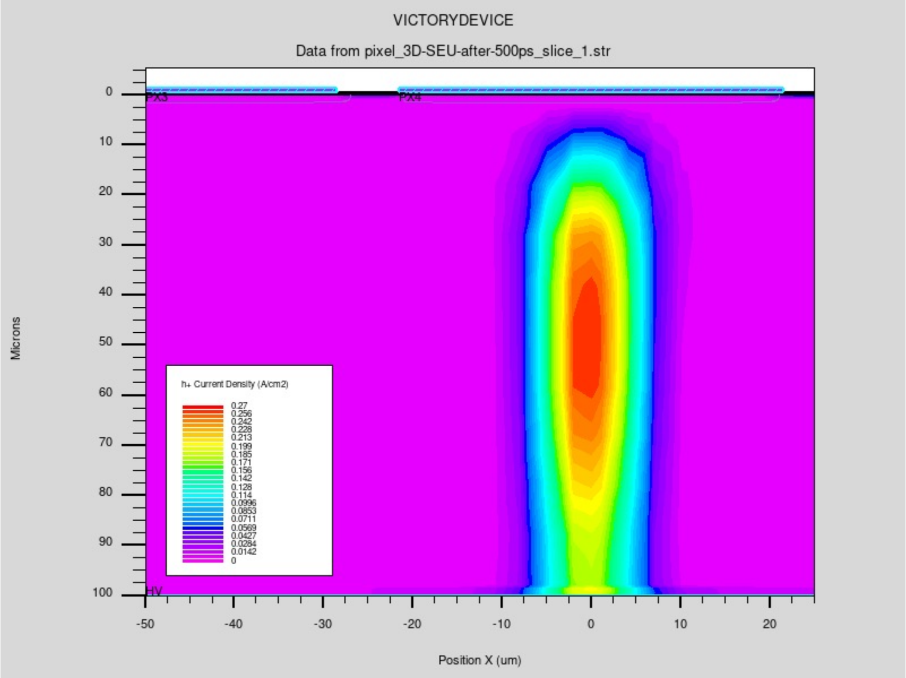
h+



Tonyplot – Current densities

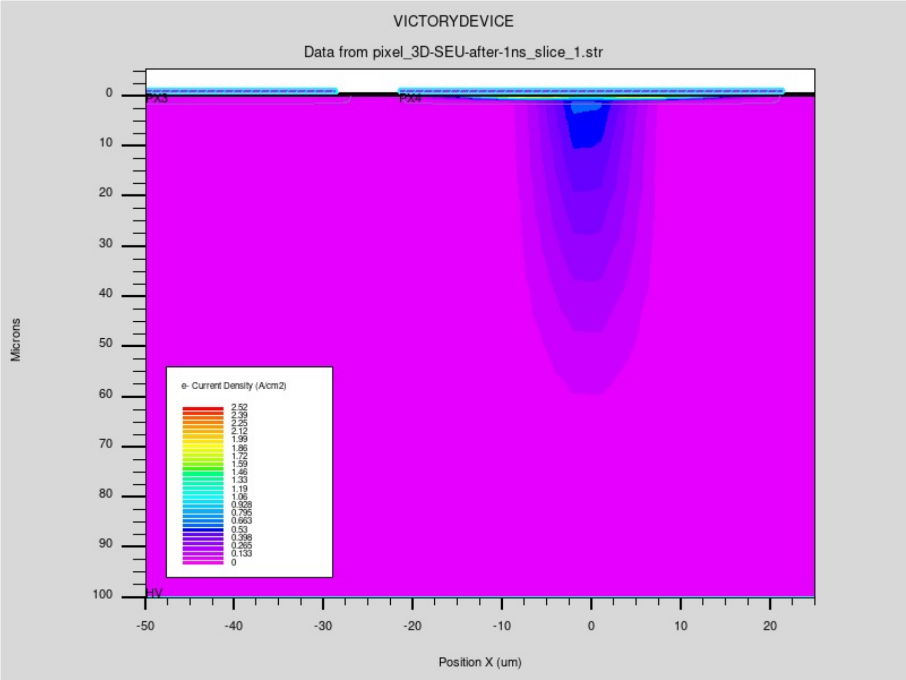
e- 500 ps after particle strike h+

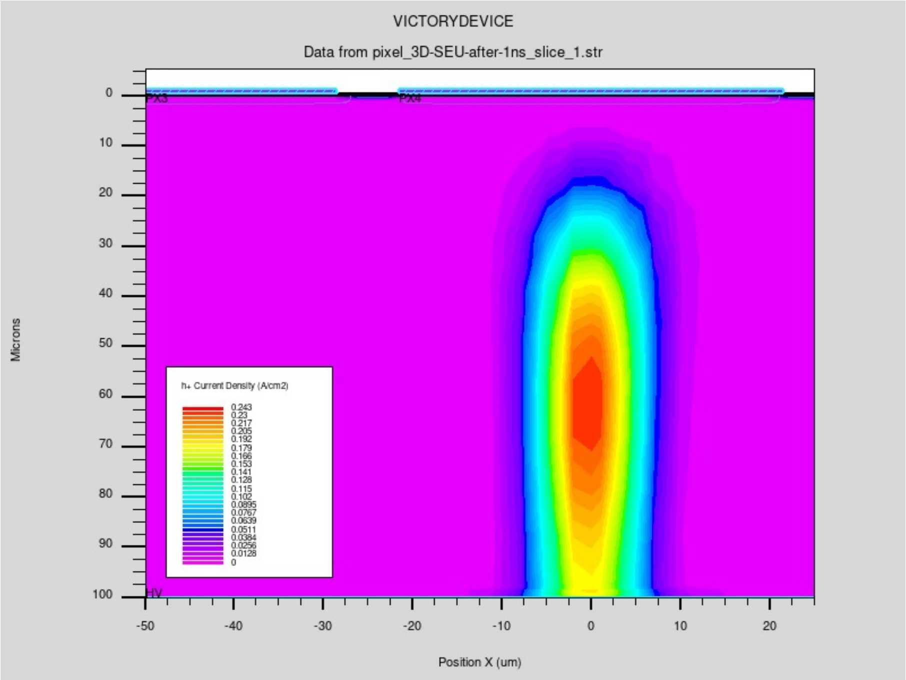




Tonyplot – Current densities

e-　　　1 ns after particle strike　　　h+

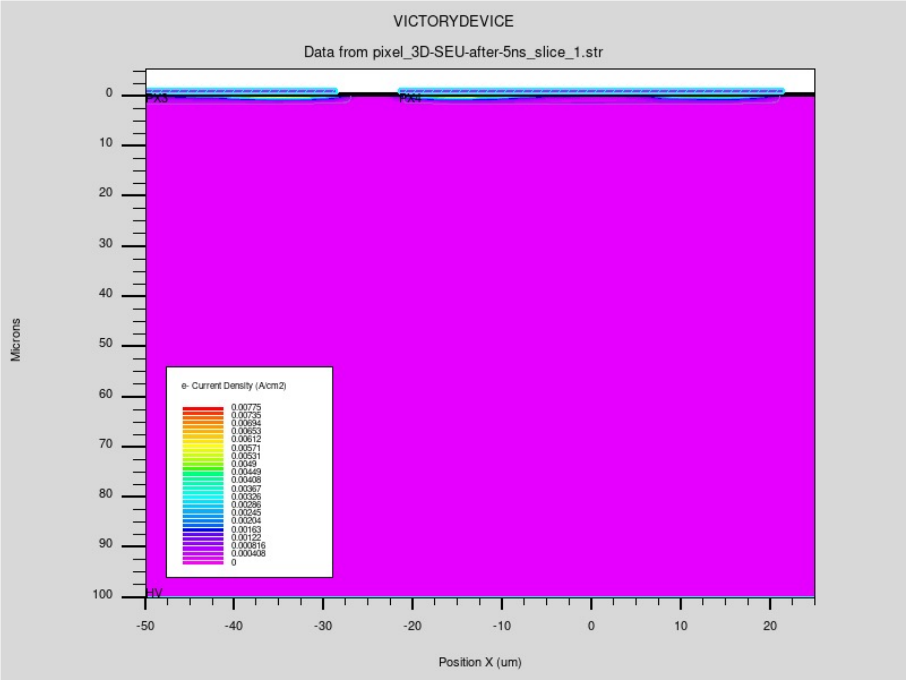




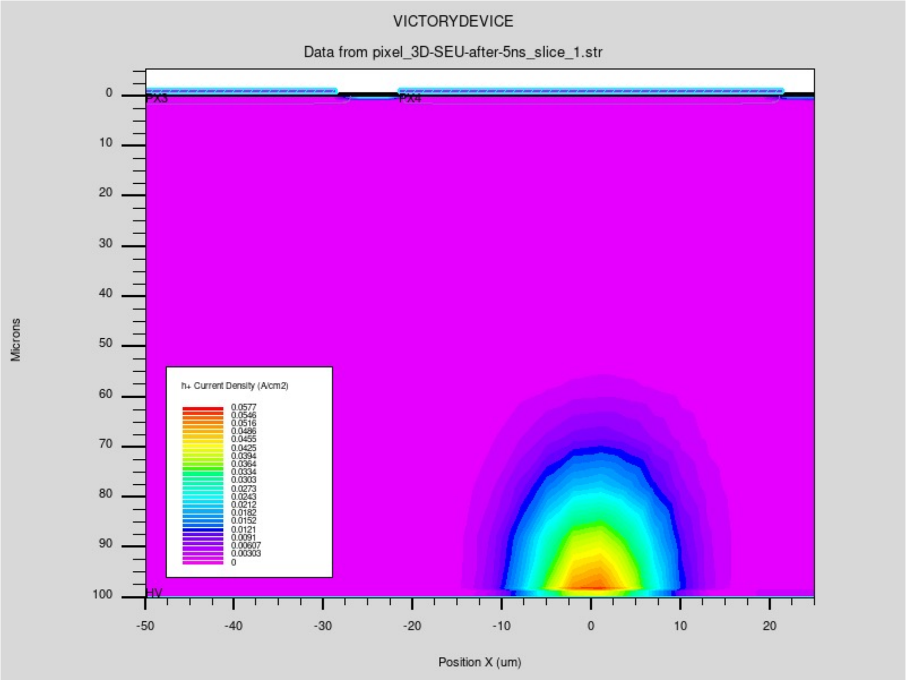
Tonyplot – Current densities

5 ns after particle strike

e-



h+



CONCLUSIONS AND OUTLOOK

Conclusions and Outlook

- TCAD is a very powerful tool for HEP silicon sensors
- You can reduce the number of submission, and so cutting time and money to get results
- Combining TCAD simulations, laboratory and testbeam data can probe fundamental quantities like electric field distribution, trapping, etc. and use them to making quantitative predictions, even after heavy irradiation
- A solid knowledge of semiconductor physics, and good data inputs are recommended to fully exploit TCAD simulations
- If you are interested in working with TCAD simulations, feel free to contact me: marco.bomben@cern.ch


https://indico.in2p3.fr/e/simdet2020
SIMDET 2020
4th school on silicon detectors simulation

THANK YOU!