Simulation Standard

Engineered Excellence

A Journal for Process and Device Engineers

2024 TCAD Baseline Release

Section 1: Process Simulation – New Features in 2024 Baseline Release

2024 baseline release of Victory Process includes major improvements and extensions of the following modules:

- Geometrical etching and deposition module
- Physical etching and deposition module
- Stress module
- Annealing module

New solid modeling capabilities are available which enable users to add more experiment-based structure information into the simulation flow. You can either insert measured surface roughness data into your simulation flow or you can initialize a 2D simulation from a TEM image of the structure or you can add TEM related shapes in your simulation. Moreover, a new engine for physical etching and deposition simulation is added which allows to simulate feature scale particle transport with high accuracy and high performance even when the sticking or reaction probability of the particles is very low. The accuracy of stress related simulations for challenging mesh configurations like high element anisotropy is considerably improved by introducing additional numerical techniques for FEM analysis which are the node based schemes (NS-FEM), the mixed node based / element based scheme (α-FEM) and the stabilized node based scheme (SNS-FEM). Besides, the advanced stress simulation workflow is unified for 2D and 3D. It is recommended to run advanced stress analysis after manually optimizing the stress simulation mesh within Victory Mesh even when the structure is generated by Victory Process to obtain the best possible simulation performance and accuracy. Note also, that 2D and 3D structures created by solid modeling capabilities of Victory Mesh and re-meshed within Victory Mesh can be used as input for stress analysis in Victory Process. Related to doping diffusion, the 2024 baseline version of Victory Process introduces a new set of diffusion models summarized under the model framework CMOS, with the objective to predict diffusion related effects relevant in advanced CMOS technology. Since this CMOS model framework introduced quite a bit of complexity from the numerical point of view major effort has been put into optimizing the numerical engines which are used to solve doping diffusion related equation systems.

1.1 Adding Measured Surface Roughness Data To the Simulation Flow

When analyzing a deposited layer, variations in layer thickness (surface roughness) are often of interest. Measurement techniques like scanning electron microscopy (SEM) can provide a layer thickness profile where the layer thickness t is a function of the substrate position t=f(x,y). The base-

INSIDE

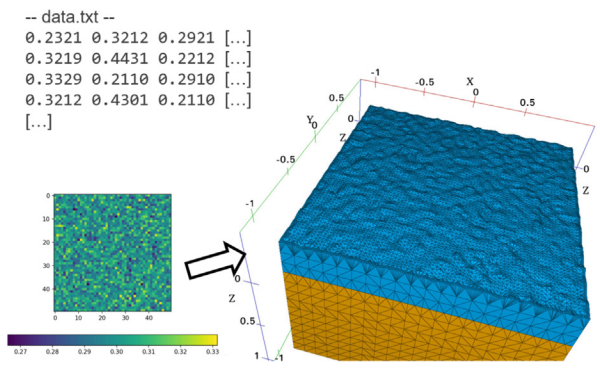


Figure 1-1: Surface roughness stored in a data file “data.txt”.

line release 2024 introduces a method to deposit layers with a given surface roughness profile f(x,y) with the *thickness modulation* feature of geometrical `DEPOSIT`.

Surface roughness data is typically available in a spreadsheet or matrix with N rows and M columns. This 2D dataset describes f(x,y) on an underlying equidistant grid. Figure 1-1 shows how layer thickness f(x,y) is described by a file containing N x M data entries. The heatmap visualizes the data with different colors for different layer thickness.

A data file “data.txt” describes layer thickness within the so-called *field of view* (FOV). FOV is a rectangle with left bottom corner LB and right top corner RT. The N x M data matrix together with FOV defines the surface roughness as a function of position *t=f(x,y)*. This is achieved by mapping the thickness profile given by the N x M data file to the FOV (Figure 1-2). The starting point (first entry in first row) is mapped to LB = (x1, y1) and the final entry in final row is mapped to RT = (x2,y2).

`DEPOSIT` with surface roughness is available with the `DIRECTIONAL` geometrical deposition model only. The path to the data file containing N x M data is provided via the parameter `THICKNESS.MODULATION.INFILE` of the `DEPOSIT` statement. The FOV is specified via the parameters `THICKNESS.MODULATION.INFILE.LB` and `THICKNESS.MODULATION.INFILE.RT`. The following example shows how you can deposit an oxide layer with thickness=1 μm that has some additional surface roughness.

```
#Deposit with thickness modulation from data.txt
DEPOSIT DIRECTIONAL MATERIAL=”oxide” \
  THICKNESS=1 \
    THICKNESS.MODULATION.INFILE=”data.txt” \
    THICKNESS.MODULATION.INFILE.LB=”-1,-1” \
    THICKNESS.MODULATION.INFILE.RT=“ 1, 1“
```

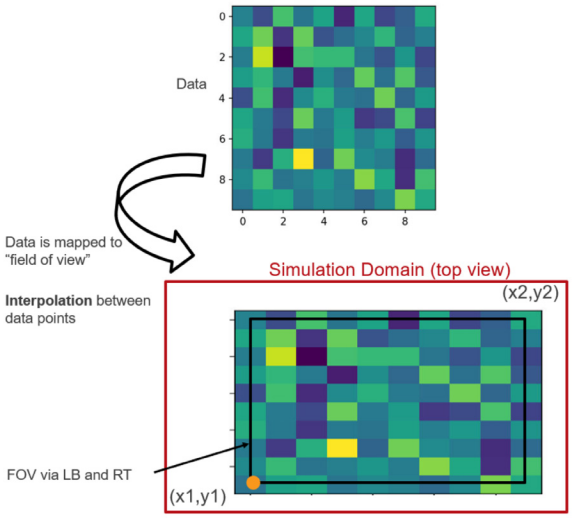


Figure 1-2: Field of view mapping with thickness modulation from file.

The locally deposited layer thickness is

$$t(x,y)=THICKNESS+f(x,y),$$
$$with\ (x,y)\ within\ FOV,$$

The layer thickness is modulated by the data matrix defining f(x,y). Hence, this technique is called thickness modulation.

For (x,y) positions outside the FOV, extension symmetry conditions are applied. Victory Process’ default behavior is to apply symmetric extension (Figure 1-3 left). You can also request periodic extension by setting the flag `THICKNESS.MODULATION.INFILE.PERIODICEXTENSION` (Figure 1-3 right). As demonstrated in Figure 1-3, `DEPOSIT` with thickness modulation is not limited to surface roughness, you can generate any structure like such L-shaped features.

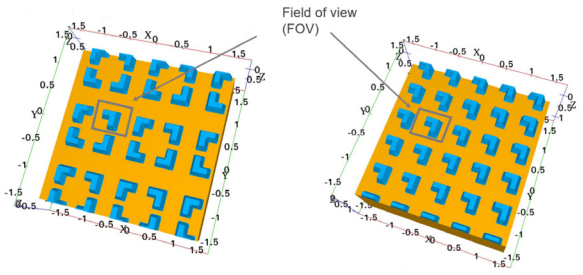


Figure 1-3: Field of view extension.

N x M data is a discrete description of layer thickness. To generate a continuous function f(x,y) for all points (x,y) be-tween the provided data points, Victory Process employs interpolation. The default technique is bilinear interpolation (Figure 1-4 left). You can also request bicubic interpolation (Figure 1-4 right) by setting the flag `THICKNESS.MODULATION.INFILE.BICUBICINTERPOLATION`.

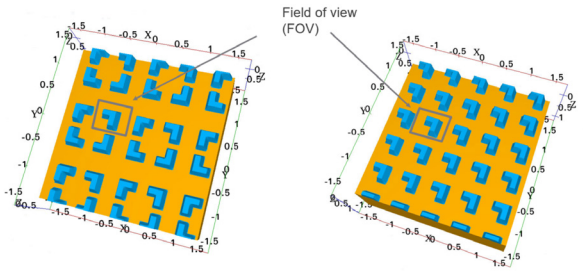


Figure 1-4: Interpolation techniques for thickness values between data values.

1.2 Create 2D Structure Directly from TEM Measurements

When analyzing a multilayer structure with transmission electron microscopy (TEM) you might extract data describing the thickness of the individual layers. The baseline release 2024 introduces a method to deposit 2D structures precisely following such shapes obtained by TEM. To illustrate, we consider a structure consisting of a substrate and three deposited layers (Figure 1-5).

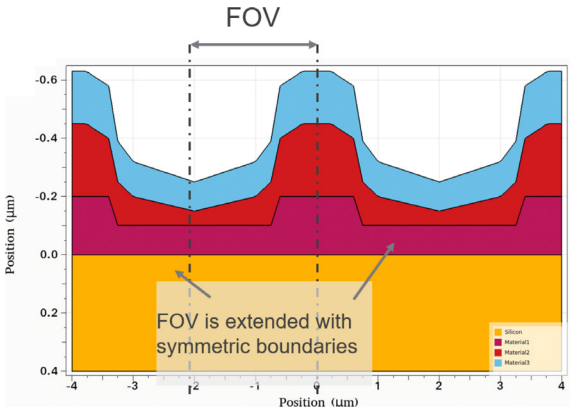


Figure 1-5 Structure generated from a ".tem" file.

Typically, a TEM data extraction tool provides a file with the following content.

```
--thickness.tem--
#TEM file with thickness data for 3 layers
(comment line)
3 6 1.0
-2.0000000 0.10 0.05 0.1
-1.0000000 0.10 0.1 0.12
-0.7500000 0.10 0.15 0.14
#-0.650000 0.20 0.20 0.16 (comment line)
-0.6000000 0.20 0.20 0.18
-0.2310000 0.20 0.25 0.18
0.0000000  0.20 0.25 0.18
```

The file extension ".tem" indicates that the data originates from a TEM measurement. The first line is a comment de-scribing where the data comes from, and the second line is the header line ("3 6 1.0"), where the first number (here 3) refers to the number of layers, the second number (here 6) is the number of rows that will follow, and the third number (here 1.0) is the bottom z coordinate of the bounding box.

The actual data section starts after this header. The first column is the x (or y – in YZ simulation mode) coordinate of the structure. The second column represents the thickness of the first layer, the third column the thickness of the second layer, etc. Hence, the first column defines spatial discretization.

Victory Process allows you to reconstruct the geometry described by the ".tem" file by multiple consecutive `DEPOSIT DIRECTIONAL` steps, whereby each `DEPOSIT` step makes use of the thickness modulation feature.

```
# Pick column 1 for first layer
DEPOSIT DIRECTIONAL MATERIAL="material1"  \
    THICKNESS=0.0 \
      THICKNESS.MODULATION.INFILE="thickness.tem" \
      THICKNESS.MODULATION.INFILE.ZCOLUMN=1

# Pick column 2 for second layer
DEPOSIT DIRECTIONAL MATERIAL="material2" \
    THICKNESS=0.0 \
      THICKNESS.MODULATION.INFILE="thickness.tem" \
      THICKNESS.MODULATION.INFILE.ZCOLUMN=2

# Pick column 3 for third layer
DEPOSIT DIRECTIONAL MATERIAL="material3" \
    THICKNESS=0.0 \
      THICKNESS.MODULATION.INFILE="thickness.tem" \
      THICKNESS.MODULATION.INFILE.ZCOLUMN=3
```

In each `DEPOSIT` step, `THICKNESS.MODULATION.INFILE.ZCOLUMN` picks the thicknesses of one of the 3 layers described in "thickness.tem". Based on the information in the ".tem" file, Victory Process automatically constructs a thickness modulation function f(x) with the final deposition thickness

$$t(x)=THICKNESS+f(x),$$
$$with\ (x,y)\ \ within\ FOV.$$

The first and the last value define the field of view (FOV). If the simulation domain is larger than FOV, the FOV is extended with symmetric (default) or periodic (enabled via `THICKNESS.MODULATION.INFILE.PERIODICEXTENSION`) boundary conditions. Note that in Figure 1-5 symmetric boundary conditions are applied.

Sometimes the layer thickness is not directly provided in the ".tem" file. Instead, the absolute z coordinate of a layer are given in the respective columns in the ".tem" file. To account for that, you can specify the flag `THICKNESS.MODULATION.INFILE.ABSOLUTEZ`, like in

```
#Deposit a HfO2 layer with the upper edge \
provided as absolute z coordinates.

DEPOSIT DIRECTIONAL MATERIAL="hfo2" \
    THICKNESS=0.0 \
      THICKNESS.MODULATION.INFILE="hfo2.tem" \
      THICKNESS.MODULATION.INFILE.ZCOLUMN=1 \
      THICKNESS.MODULATION.INFILE.ABSOLUTEZ
```

Furthermore, if your data represents y-z data (in a 3D simulation exercise) you must specify the flag `THICKNESS.MODULATION.INFILE.YZDATA`.

1.3 Top-down Flux Engine for Physical Etch and Deposition

Victory Process' most advanced etching and deposition models often employ ballistic feature scale reactant transport methodology (ballistic flux), as illustrated in Figure 1-6 for the case of fluorocarbon compound particles. In real world, such particles originate from a particle source in the reaction chamber, but since Victory Process is a feature scale simulator the incoming particles are assumed to originate from a source plane located slightly above the simulated structure. After particles are emit-ted from the source, they hit the structure and interact with the surface. The goal is to perform predictive etch-ing or deposition simulation with models accurately de-scribing the particle propagation as well as the physics and chemistry of the surface interactions. This allows the simulations to capture non-ideal effects like micro-trenching, facet formation, or bowing.

The baseline release 2024 introduces a major new mode for ballistic flux simulation, called the Monte-Carlo top-down flux engine. To trace the trajectory of the particles contributing to etching or deposition, the top-down flux engine employs ray tracing, a highly optimized technique that is well-known from 3D computer graphics. The high computational efficiency of ray tracing is beneficial for ballistic flux calculation because trajectories of numerous particles must be considered to achieve accurate etching and deposition simulation results.

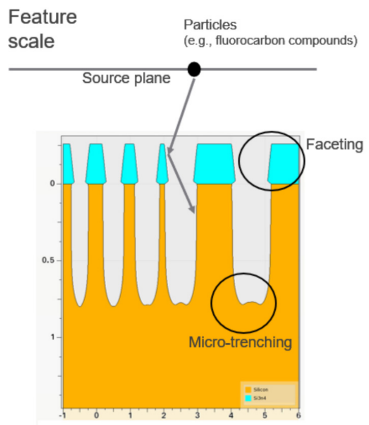


Figure 1-6: Ballistic flux simulation.

The new top-down (TD) flux engine is available in addition to Victory Process' traditional bottom-up flux (BU) engine. Table 1-1 summarizes the capabilities and differences of BU and TD flux engines. Most importantly, TD flux engine significantly improves performance and accuracy of 3D simulations with many reflections of particles at the surface of the geometry.

Bottom-up flux engine	Top-down flux engine
Calculate flux integral at all surface points	Based on ray tracing: Follow particle trajectory
Based on visibility between two points	Monte Carlo Method (statistical method)
High accuracy for direct flux (i.e., no reflection or scattering)	High accuracy for secondary flux (many reflection) with low sticking coefficient
Repeated specular reflection is not possible	Full support for specular reflection (including specular reflection with a diffusive component)
Supports C function for customizable models	Supports C function for customizable models
Special performance optimization for 2D	High performance in 2D and 3D
Computationally expensive for 3D	Highly parallelizable Scales well with number of cores
	Optimized diffusive reflections (important for low sticking coefficients)
	Well-suited for advanced ion milling models, due to supports of initial energy particle distribution and energy reduction during surface interaction.

Table 1-1: Comparison of bottom-up and top-down flux engine.

The top-down flux engine is selected by adding the flag `TOPDOWN` to the `FLUX` statement. The setup for primary and secondary fluxes is the same as for the BU flux engine. However, when `TOPDOWN` is enabled, additional parameters are available. Most importantly, the parameter `TOPDOWN.RAYSPERPOINT` which sets the number of particle trajectories that are calculated per surface point. This parameter considerably determines the accuracy of the simulation result. Usually, a number around 1000 is sufficient to achieve very accurate results.

Furthermore, you can configure TD flux with "special" (dedicated) primary and secondary flux models like in this deck statement example:

```
FLUX NAME="enhancedflux" TOPDOWN \
      TOPDOWN.RAYSPERPOINT=1000 \
      TOPDOWN.PRIMARY.POWERCOSINE \
      TOPDOWN.PRIMARY.POWERCOSINE.EXPONENT=100 \
      TOPDOWN.SECONDARY.SPECULAR
```

Pre-defined flux models cover a broad range of applications and employ optimized random number generation and sampling, resulting in a fastest simulation.

In the example shown here, the deck parameters `TOPDOWN.PRIMARY.POWERCOSINE` and the associated deck parameter `TOPDOWN.PRIMARY.POWERCOSINE.EXPONENT` configure the distribution of particles at the source plane. Figure 1-7 visualizes such a power cosine particle distribution. The larger the exponent the more directed the particle flux is.

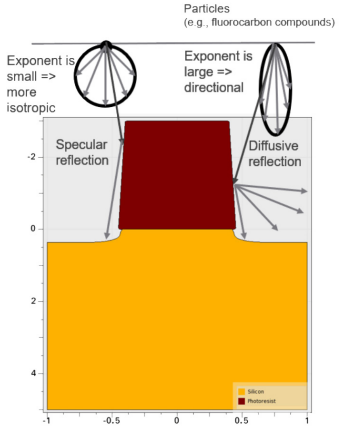


Figure 1-7: Power cosine distributions with small and large exponent and ideal reflection types.

The deck parameter `TOPDOWN.SECONDARY.SPECULAR` determines how particles are reflected when they hit the surface. In the example shown here perfectly specular (i.e., mirror-like) reflection is selected. An alternative choice would be ideal diffusive reflection also called scattering according to Lambert's cosine law. In that case the parameter `TOPDOWN.SECONDARY.DIFFUSIVE` would be specified instead. Both such types of reflections are also illustrated in Figure 1-7.

TD flux is well-suited for deposition processes which exhibit "diffusive" properties. Figure 1-8 shows a structure with lateral slits and narrow openings. Incoming particles cannot directly reach the inner regions of the lateral slits. Instead, a process of repeated diffusive reflections is needed to eventually lead to almost conformal deposition of material along the slits. With BU flux, this is only possible with very high computational effort. In contrast, due to the high computational efficiency and optimized diffusive reflections, TD flux is well-suited to simulate deposition processes of this type.

To set up such a diffusive like deposition process, the TD flux engine is configured with the parameter `TOPDOWN.SECONDARY.DIFFUSIVE`. Repeated scattering is achieved by setting a small sticking coefficient of 0.001. The sticking coefficient characterizes the probability that an incoming particle is embedded into surface and hence sticks to the surface. The following deck snippet of Victory Process demonstrates this.

```
FLUX NAME="diffusiveflux" TOPDOWN \
  TOPDOWN.RAYSPERPOINT=1000 \
  TOPDOWN.PRIMARY.POWERCOSINE \
    TOPDOWN.PRIMARY.POWERCOSINE.EXPONENT=10 \
  TOPDOWN.SECONDARY.DIFFUSIVE

TOPOGRAPHYMODEL NAME="diffusivedepo" \
  FLUX="diffusiveflux" \
  REACTION="fluxdependent"

ETCHDEPOPROPERTIES NAME="matprop" \
  MATERIAL="fancy_oxide" \
    RATE=0.01 STICKING=0.001 \
  MATERIAL="oxide"  \
    RATE=0.01 STICKING=0.001

DEPOSIT MATERIAL="fancy_oxide"
  MODEL="diffusiveDepo" \
  MATERIALPROPERTIES="matprop" \
  TIME=0.2 MAXCFL=1
```

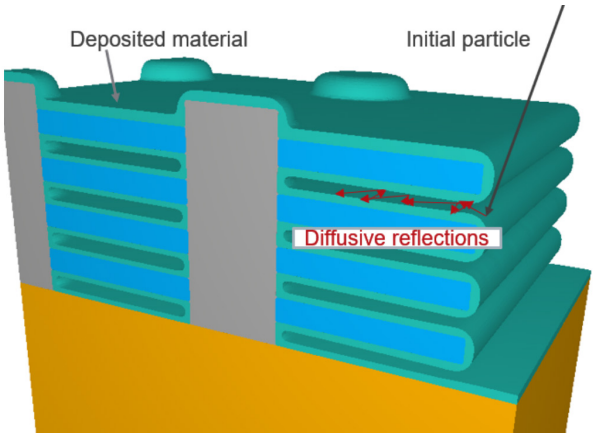


Figure 1-8: Top-down flux enables efficient diffusion like deposition models.

Note that the statements `TOPOGRAPHYMODEL`, `ETCHDEPOPROERTIES`, and `DEPOSIT` are the same for the TD and BU flux engine, only the `FLUX` statement differs.

Having the capability to efficiently simulate particles experiencing multiple reflections, the TD flux engine allows you to model the bowing effect caused by tapered (hard) mask structures in a deep reaction ion etching process. To illustrate this, Figure 1-9 shows a structure with numerous stacked layers. Many incoming etchant particles first hit the mask before hitting the layer stack. These particles are reflected at the mask and are sub-sequently primarily hitting upper parts of the trench. The physical reflection mechanism is best described by a non-ideal specular reflection where the outgoing particle trajectory is randomly distributed around the ideal specular reflec-

tion direction. A model for this is called “coned_specular_reflection” and is available in Victory Process’ open etching/deposition model library. For in-stance, the FLUX statement

```
FLUX NAME=”tdflux” TOPDOWN \
  TOPDOWN.PRIMARY.POWERCOSINE \
    TOPDOWN.PRIMARY.POWERCOSINE.EXPONENT=900 \
  SECONDARY.FUNCTION=”coned_specular_reflection” \
    SECONDARY.DEP1=”angletospecular” \
    SECONDARY.PAR1=20 \
  TOPDOWN.RAYSPERPOINT=1000
```

configures TD flux with highly directional particles coming from the source (power cosine exponent=900) and coned specular reflection with cone opening angle of 20°. The remaining etch model configuration is almost the same as the one given above, with the only exception being a sticking coefficient of 0.7 for all materials.

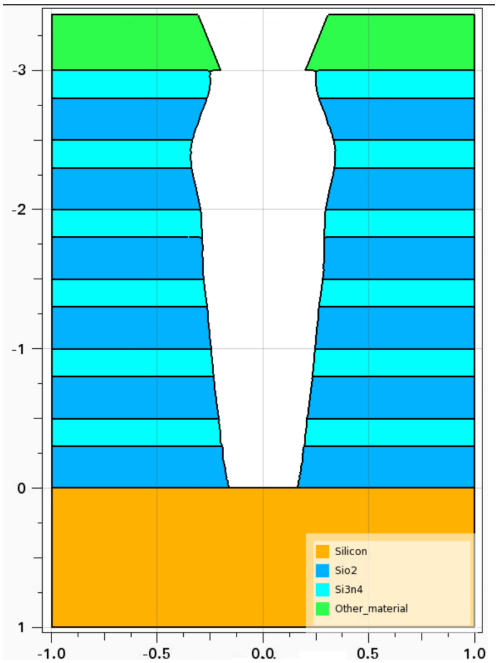


Figure 1-9: Bowing effect with top-down flux engine.

1.4 Node-based Smoothed Finite Element Method for Stress Analysis

Because of their suitability for arbitrarily complex geometries and fully automatic mesh generations, the use of tetrahedral elements is a key aspect in stress analyses for manufacturing process simulation that require seamless finite element discretization. However, in practical applications, the standard linear tetrahedral elements have many limitations and drawbacks, such as poor accuracy (overly stiff), sensitive to mesh distortion, shear/volumetric locking phenomenon. These drawbacks seriously restrict the application of the standard FEM and many numerical improvements have been developed to solve such issues.

Among these methods, the node-based smoothed finite element method (NS-FEM) is regarded as one of the most promising methods. The baseline release 2024 introduces the NS-FEM as a method of choice for stress analysis, besides the other numerical techniques which had been available in previous versions.

When extending the NS-FEM to various physical problems, the smoothing operation is carried out based on the gradient of the physical field variables (such as the acoustic pressure, temperature and displacement, magnetic vector potential and so on). In the formulation of the NS-FEM, the numerical integration procedures for the system stiffness matrix are performed based on the node-based smoothing domain which is constructed based on the elements but beyond the elements. Typically, the problem domain Ω is first discretized using N_e triangular or tetrahedral elements in the same manner as in the standard FEM. Based on the obtained background mesh, the problem domain is further subdivided into N_n non-overlapping and non-gap smoothing domains, in which N_n denote the total number of field nodes.

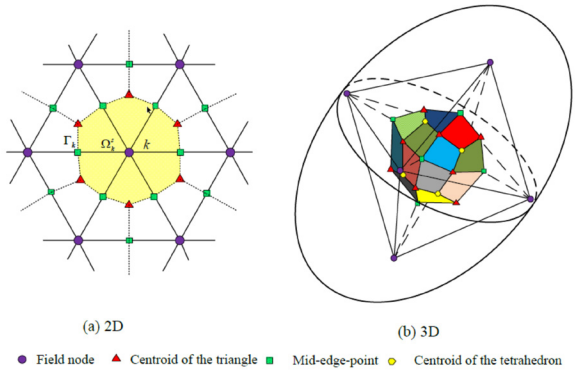


Figure 1-10: Node-based gradient smoothing operation.

Figure 1-10 shows a common process to construct the node-based smoothing domain. For an interior node k, in two-dimensional spaces, the node-based smoothing domain is constructed by linking the mid-edge points and the central points of the elements associated with the node k in order. When it comes to three-dimensional spaces, the smoothing domain for node k can be constructed by linking the mid-edge points, the centroids of surface triangles together with the central points of the tetrahedrons associated with the node in proper order.

When the standard FEM is employed to solve some common physical field problems, the field variables within each element can be obtained using the interpolation form

$$\boldsymbol{\varphi} = \sum_{i=0}^{N_p} N_i(\boldsymbol{x})\boldsymbol{\varphi_i}$$

Equation 1-1

where

- φ denotes the physical field variables (scalar: acoustic pressure, temperature; vector: displacement, magnetic vector potential);
- N_p is the number of nodes in each element;
- $N_i(x)$ is the shape function at i-th node;
- φ_i is unknown nodal field variables value.

Based on the standard Galerkin weak form, the system stiffness matrix **K** can be written in the following general form

$$\boldsymbol{K} = \int_{\Omega} (\nabla \boldsymbol{N})^T \boldsymbol{D} (\nabla \boldsymbol{N}) d\Omega = \int_{\Omega} \boldsymbol{B}^T \boldsymbol{D} \boldsymbol{B} d\Omega = \sum_{i=1}^{N_e} \int_{\Omega^e} \boldsymbol{B}^T \boldsymbol{D} \boldsymbol{B} d\Omega \quad 2$$

in which

- **B** is the general gradient matrix;
- **D** is a matrix of material constants.

Using the gradient smoothing technique and introducing the Green's divergence theorem, the gradient of the physical field variables in the node-based smoothing domain Ωs can be generally expressed as

$$\nabla \bar{\boldsymbol{\varphi}} = \frac{1}{A_k} \int_{\Omega^s} \nabla \boldsymbol{\varphi} d\Omega = \frac{1}{A_k} \int_{\Gamma^s} \boldsymbol{\varphi} \cdot \boldsymbol{n} d\Omega = \sum_{i \in M_k} \bar{\boldsymbol{B}}_i \boldsymbol{\varphi}_i$$

Equation 1-3

where

- $\nabla \bar{\boldsymbol{\varphi}}$ represents the smoothed gradient of the field variables;
- A_k denotes the area or the volume of the smoothing domain;
- Γ^s is the boundary of the node-based smoothing domain;
- M_k denotes a set containing all nodes located in the influence domain of node k.

It should be noted here that Equation 1-3 is just a general form of the gradient items, and the specific form of a certain physical problem may be slightly different from this. As the Green's divergence theorem is introduced, the area integration over the smoothing domain is converted into the line integral along Γs. Replacing the compatible gradient component shown in Equation 1-1 with the smoothed gradient, the smoothed system stiffness matrix can be further explained as

$$\bar{\boldsymbol{K}} = \sum_{k=1}^{N_n} \bar{\boldsymbol{K}}_{\boldsymbol{k}} = \int_{\Omega^s} (\bar{\boldsymbol{B}}_{\boldsymbol{k}})^T \boldsymbol{D} \bar{\boldsymbol{B}}_{\boldsymbol{k}} d\Omega$$

Equation 1-4

This formulation is referred to as node-based smoothed finite element method (NS-FEM) as the system stiffness matrix is calculated based on the smoothing domain associated with nodes. Studies have shown that the NS-FEM is well immune from volumetric locking and shear locking. In addition, as the field gradients are obtained directly through the shape functions, i.e., no coordinate transformations are involved, the NS-FEM performs well even when severely distorted elements are employed or in other words it is less sensitive to a mesh with bad aspect ratio elements. However, the NS-FEM possesses the upper bound property in strain energy that makes the solution overly soft. The calculations required to form the stiffness matrix of the NS-FEM are similar to those for the standard FEM, but there is greater coupling between degrees of freedom in the assembled stiffness matrix of the entire mesh. Even though this greater degree of coupling leads to more nonzero entries in the assembled matrix, the improvements in accuracy outweigh the additional time required for matrix factorization.

Although the node-based smoothed finite element method (NS-FEM) possesses many superior properties such as locking-free and mesh insensitivity, it suffers from instability that is mainly caused by the "overly soft" property of the NS-FEM model. To cure the instability of the node-based smoothed finite element method, two stabilization schemes are employed to strengthen the system stiffness matrix shown in Equation 1-4.

α-FEM

A class of numerical improvements is formulated by combining the "overly soft" NS-FEM with the "overly stiff" FEM, which is called the alpha finite element method (α-FEM).

To this end, a parameter `ALPHA` is introduced in the `STRESS` statement:

```
STRESS … ALPHA=<value>
```

With this, the stiffness matrix is combined using the `ALPHA` as weighting factor such that

$$K^{\alpha} = \alpha \boldsymbol{K} + (1 - \alpha) \bar{\boldsymbol{K}}$$

Equation 1-5

Thus, the formulation collapses to the standard FEM for alpha=1 while the formulation collapses to the NS-FEM for alpha=0. This method provides a very accurate numerical solution with a proper parameter `ALPHA`. However, it is still an unsolved problem, how to obtain an optimal parameter value, as both the nature of the problem and the size of the mesh discretization will have great influence on the parameter value.

Stabilized Node-based Method (SNS-FEM)

Recently, a stable node-based smoothed finite element method (SNS-FEM) without any uncertain parameter has been proposed, where the simplest linear triangular and tetrahedral elements are employed to discretize the problem domain and the node-based smoothing domain is then further constructed. Unlike the original NS-FEM, the gradient variances of the field variables are taken into account to construct the stable items to strengthen the system stiffness.

As described above, the gradient of the field variables in each smoothing domain is constant, that is, gradient changes over the smoothing domain are ignored. Thus, gradient variances of the field variables are considered to construct the stable items. Generally, the smoothing domain is a polygon (2D) or polyhedron (3D) that can be approximated as an ellipse or elliptical sphere domain with the same area or volume. We show a 2D case here for simplicity while the 3D case is the same (only add z-direction). The equivalent ellipse is then further divided into four sub-domains equally. And based on the obtained sub-domains, four points located in x-axis, y-axis with the distance rx, ry to node k are chosen to be the integration points *gi* (i=1,2,3,4). Assuming that the gradient of the field variables in the smoothing domain is continuous and derivable at the first order, thus, the Taylor expansion of the gradient at node k can be expressed as

$$\nabla\boldsymbol{\varphi} = (\nabla\boldsymbol{\varphi})_k + \frac{\partial(\nabla\boldsymbol{\varphi})}{\partial x}(x - x_k) + \frac{\partial(\nabla\boldsymbol{\varphi})}{\partial y}(y - y_k)$$

Equation 1-6

After this, the gradient items at each integration points can be obtained as

$$\begin{aligned}
(\nabla\boldsymbol{\varphi})_{k_{s1}} &= (\nabla\boldsymbol{\varphi})_{k_s} + \frac{\partial(\nabla\boldsymbol{\varphi})}{\partial x} r_x \\
(\nabla\boldsymbol{\varphi})_{k_{s2}} &= (\nabla\boldsymbol{\varphi})_{k_s} - \frac{\partial(\nabla\boldsymbol{\varphi})}{\partial x} r_x \\
(\nabla\boldsymbol{\varphi})_{k_{s3}} &= (\nabla\boldsymbol{\varphi})_{k_s} + \frac{\partial(\nabla\boldsymbol{\varphi})}{\partial y} r_y \\
(\nabla\boldsymbol{\varphi})_{k_{s4}} &= (\nabla\boldsymbol{\varphi})_{k_s} - \frac{\partial(\nabla\boldsymbol{\varphi})}{\partial y} r_y
\end{aligned}$$

Equation 1-7

By substituting the above equations into the smoothed Galerkin weak form, the smoothed stiffness matrix over the smoothing domain can be modified as

$$\hat{\boldsymbol{K}}_k = \bar{\boldsymbol{K}}_k + \frac{A_k}{2}(\bar{\boldsymbol{B}}_{kx})^T \boldsymbol{D} \bar{\boldsymbol{B}}_{kx} + \frac{A_k}{2}(\bar{\boldsymbol{B}}_{ky})^T \boldsymbol{D} \bar{\boldsymbol{B}}_{ky}$$

Equation 1-8

Where $\bar{\boldsymbol{B}}_{kx} = \frac{\partial(\nabla\boldsymbol{\varphi})}{\partial x} r_x$

Obviously, the system stiffness matrix has been strengthened by stable terms introduced by the smoothed gradient expansion of the strain at the node. The additional terms in the stabilization are necessarily positive for nonzero strain and thus provide additional correctness insurance. Moreover, in contrast to other stabilized methods, the constants associated with the additional terms do not include a tuning parameter.

To invoke the SNS-FEM in the 2024 baseline version of Victory Process, you need to set the parameter ALPHA=0 in `STRESS` statement.

Example – Cantilever Bending

A standard benchmark is a cantilever beam with dimension 10 x 1 x 0.2um (L=10, b=1.0, h=0.2) and vertical force in one end (P=0.02 dyne). The analytic solution is given by

$$\delta = \frac{PL^3}{3EI}$$

Equation 1-9

For the given Young's modulus E=1000, the analytic solution is 1.

A corresponding 2D Victory Process simulation deck is:

```
INIT MATERIAL="matA" DEPTH=0.2 GASHEIGHT=0.4 \
    FROM=0 TO=10 FLOW.DIM=2d_xz RESOLUTION=0.05

MATERIAL NAME="matA" \
  STRESS.PARAMETER="young.modulus" \
    STRESS.PARAMETERVALUE=10e11 \
  STRESS.PARAMETER="poisson.ratio" \
    STRESS.PARAMETERVALUE=0.0

METHOD STRESS.ANISO=off \
    STRESS.FORM2D="planestress" \
     STRESS.LARGE_DEFORM=off

STRESSBOUNDARYCONDITIONS NAME="case_1" \
                 Z.FREE XM.FREE \
                 SPOINT="0, 0" \
                   SEARCHTOLERANCE=0.03 \
                   FORCE="0, 0.02"

STRESS BC="case_1" TRANSFER.RESULTS ALPHA=0
EXPORT DEFORMED STRUCTURE="planestress.str"
```

As shown in Table 1-2, the result for the standard FEM is overly stiff while the result for the nodal smooth-ing NS-FEM is overly flexible as expected. The accuracy of the solution has been significantly improved by the α-FEM with `ALPHA`=0.3. However, it is hard to know which α is good enough in advance. The stabilized node-based scheme (SNS-FEM) provides the exact solution without any user parameter.

	2D	3D
FEM (alpha=1)	0.82	0.82
NS-FEM (alpha=1e-05)	1.12	1.12
α-FEM (alpha=0.3)	1.01	1.01
SNS-FEM (alpha=0)	1.00	1.00

Table1-2: Tip displacement of cantilever beam (analytical solution = 1.00).

Example – 2D TFT Bending

To demonstrate the instability associated with the pure node-based scheme (NS-FEM), we simulate an IGZO TFT on a flexible substrate. In this example the target structure is composed of six different material regions including IGZO/Cr (gate metal)/PI. Since the critical point in thin film problems usually happens to be the most brittle layer that fails to resist to the applied bending stress, SiO2 buffer layer between film and PI substrate is introduced to resist such stress induced failure stack as shown in Figure 1-11.

The mechanical properties used for this simulation are listed in Table 1-3. Since instability is often observed in largely deformed structure, we apply prescribed rotations (30 degree) at both ends to cause substantial deformation.

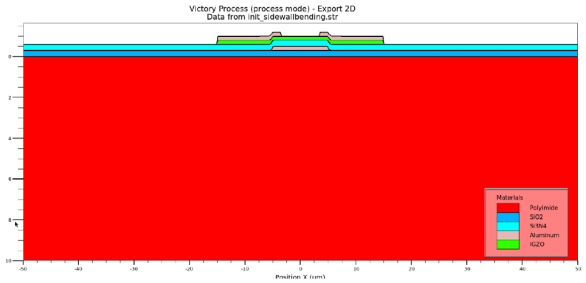


Figure 1-11: IGZO TFT on a flexible substrate.

	Young's modulus (dyne/cm^2)	Poisson ratio	thickness(um)
IGZO	1.37e12	0.36	0.2
Si3N4	2.5e12	0.23	0.3
Mo	3.3e12	0.38	0.2
Cr	2.79e12	0.21	0.2
SiO2	0.7e12	0.17	buffer
PI	0.29e12	0.34	10um

Table 1-3: Material properties and geometry related simulation input.

Here, four different formulations are tested:

- standard FEM (alpha=1),
- NS-FEM (alpha=1e-05),
- α-FEM (alpha=0.05), and
- SNS-FEM (alpha=0).

Figure 1-12 shows strain contours (XX component) of all four cases. The wavy deformations on top corner surfaces in NS-FEM are observed due to the instability of the numerical scheme. The strain contour is also unreasonable in pure node-based scheme (NS-FEM). As you can see, this instability can be resolved by either α-FEM (with a small ALPHA) or SNS-FEM.

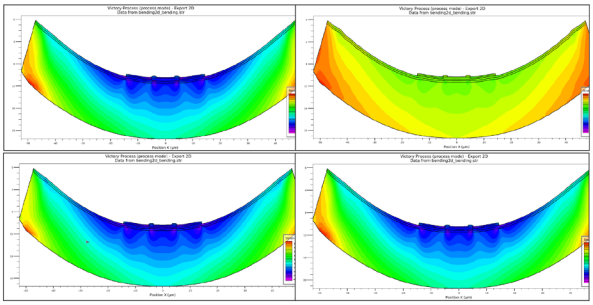


Figure 1-12: Strain contour (top left: alpha=1, top right: alpha=1e-05, bottom left: alpha=0.05, bottom right: alpha=0).

Example – Wafer Warpage

A 6-in wafer (without a flat) is modeled. The wafer is a planar disk made of Si substrate with oxide thin film on top. The details of the simulated structure are as follows:

Diameter = 6 inch (152.4 mm)
Film thickness = 0.4012 um
Substrate thickness = 675 um
Substrate Young's modulus = 150e10 dyne/cm2
Substrate Poisson's ratio = 0.27
Film Young's modulus = 150e10 dyne/cm2 (150 GPa)
Film Poisson's ratio = 0.27
Film intrinsic stress = 3.83073e+9 dyne/cm2

We fix the two center points, top and bottom, and apply biaxial film stress. The wafer warps into a bowl shape, which is the most observed wafer shape during wafer processing, by applying the same biaxial stress (SX=SY). In this case, we can analytically calculate the maximum vertical displacement (bow) from the following Stoney's formula:

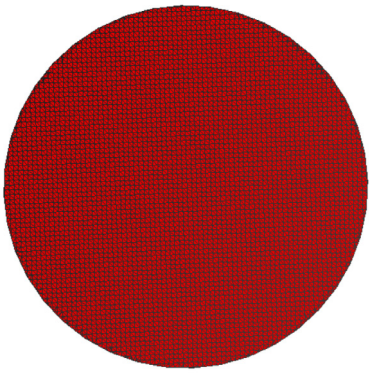
$$\delta_{max} = \frac{6D^2\sigma_f t_f(1-\nu_s)}{8E_s t_s^2} \qquad (10)$$

For the given data, the analytic bow is 28.6 um.

To perform the simulation, a conformal mesh is generated by Victory Mesh for FEM analysis (Figure 1-13). That mesh is highly anisotropic as shown in the following summary reported by Victory Process:

Element aspect ratio in material region film
average = 1:6727 *maximum = 1:7034*

Element aspect ratio in material region silicon
average = 1:664 *maximum = 1:4716*



Fiure 1-13: Conformal mesh for wafer warpage simulation.

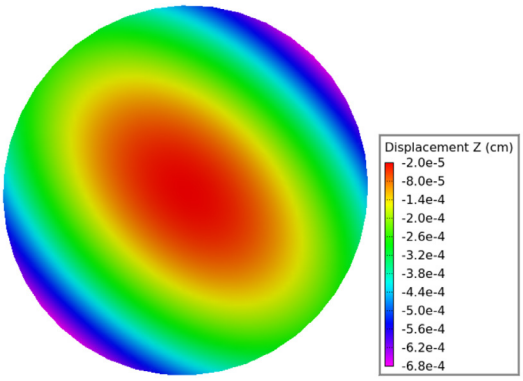


Figure 1-14: Contour of vertical displacement from the standard FEM (alpha=1).

As shown in Figure 1-14 the standard FEM gives qualitatively unsymmetric and quantitatively very stiff bow, only 6.8 um compared to 28.6 um from Stoney's formula, due to shear locking from the thin elements.

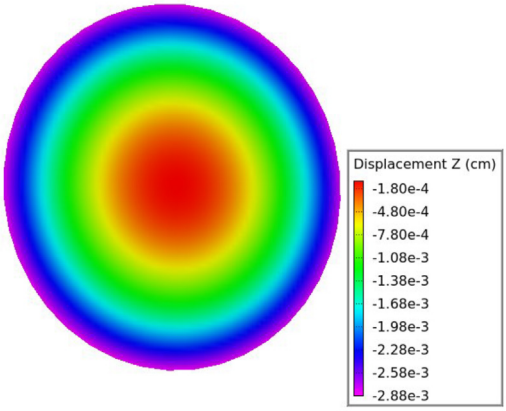


Figure 1-15: Contour of vertical displacement from SNS-FEM (alpha=0).

The stabilized node-based method (SNS-FEM) dramatically improves the situation (Figure 1-15). The contour is symmetric, and the bow is now 28.8 um that is close to the analytic solution. By applying skew-symmetric biaxial stress (SY=-SX), a saddle shape warpage is generated which is also handled well by the stabilized node-based method (SNS-FEM) as shown in Figure 1-16.

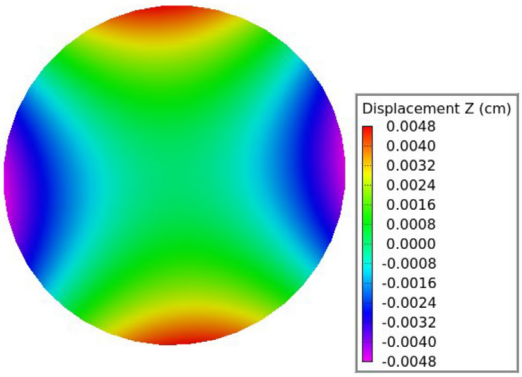


Figure 1-16: Saddle shape warpage from SNS-FEM (alpha=0).

1.5 Unified 2D and 3D Workflow for Advanced Stress Simulation Analysis

The typical simulation flow of a stress simulation in Victory Process consists of two steps:

1. Building the structure, generally using `INIT`, `ETCH` and `DEPOSIT` statements
2. Computing the stress using stress related statements such as `STRESS`, `STRESSBOUNDARYCONDITIONS` and `SETSTRESS` etc.

The automatically created stress mesh (unstructured triangular or tetrahedral mesh) of Victory Process in this simulation flow, is often suboptimal, i.e., too coarse, too fine, or its elements aspect ratio is too high for successful stress computation.

Therefore, it is recommended to add a re-mesh step using Victory Mesh in between the structure building and the stress simulation and run the stress computation in stress-Only mode. The recommend simulation flow for a stress simulation in the baseline release 2024 of Victory Process is:

```
#structure building
go victoryprocess
INIT …
…
SAVE NAME=”structure”

#re-meshing
go victorypmesh
LOAD IN=”structure”
…
SAVE OUT=”remeshed_structure.str”

#computing stress
go victoryprocess
LOAD NAME=”remeshed_structure.str” LOAD-
STRESSMESH   STRESSBOUNDARYCONDITIONS …
…
STRESS …
EXPORT STRUCTURE=”stressed_structure.str”
```

Re-Meshing

The goal of remeshing is to create a good mesh for the stress simulation (FEM analysis). A good mesh is essential to get accurate results and to ensure that the FEM solver can converge. Here a case study on a STI structure with thin layers on top (Figure 1-17) is presented. The thin layers of the structure cause the creation of high aspect ratio elements in the default stress mesh with the consequence that the stress solver fails to solve on the default stress mesh for certain loading and boundary conditions. By manually controlling remeshing and refinement settings a better mesh for stress simulation can be created as demonstrated in the following. In Figure 1-18, the elements of the stress mesh in critical areas are shown as we progress through the remeshing steps.

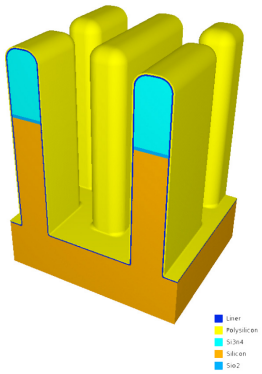


Figure1- 17 Full structure for stress simulation.

Typical commands used in Victory Mesh and guidelines to create a good stress mesh are the following:

After loading a structure created by Victory Process (in process mode) into Victory Mesh, start by cleaning artefacts/small elements using

```
OPTIMIZE MIN.SIZE=1e-3
```

This removes any features smaller than the specified `MIN.SIZE` (in um). This is recommended for complex structures created in a process mode simulation, especially ones which use a coarse `RESOLUTION`. In this cleaning step. `MIN.SIZE` should be of the order of

```
MIN.SIZE <~ RESOLUTION / 4 ^ (MESHDEPTH – 1)
```

with `RESOLUTION` and `MESHDEPTH` as set in INIT statement of the Victory Process simulation. When you perform a process simulation using cell mode of Victory Process this step can usually be omitted.

Next, go for the main re-mesh by

```
REMESH DELAUNAY MANIFOLD
```

This creates a minimal Delaunay mesh for the structure. The optional parameter `MANIFOLD` enables advanced topological-based feature segmentation using a low number of elements to represent the surface and interfaces in the structure.

However, this mesh would be too coarse for stress simulation (see Figure 1-18b). Therefore, in the application case presented here, the remeshing continues with two refinement steps. First, a general refinement for all elements is applied using

```
REFINE REGION="*" MAX.SIZE=0.0025
```

After doing so, the size of all elements in the mesh is below the specified MAX.SIZE (see Figure 1-18c). By applying this step curved interfaces in the structure are better approximated and hence they become smoother. The choice of the value provided in MAX.SIZE is usually related to the size (radius) of curved areas in the structure. The refinement is finalized by explicitly refining thin layers in the structure, with a statement like

```
REFINE REGION="poly,liner" \
  MAX.SIZE=0.0005 CONSTRAINED
```

Hereby you ensure that you have mesh points within the thin layers. Those mesh points are required for numerical reasons by the stress simulation engine. Usually, you should have at least 3 mesh points in thickness direction of thin layers (see Figure 1-18d). Note that the `CONSTRAINED` refinement does not move interfaces, i.e., material boundaries and consequently, all statements following

```
REFINE CONSTRAINED
```

must also use the `CONSTRAINED` parameter.

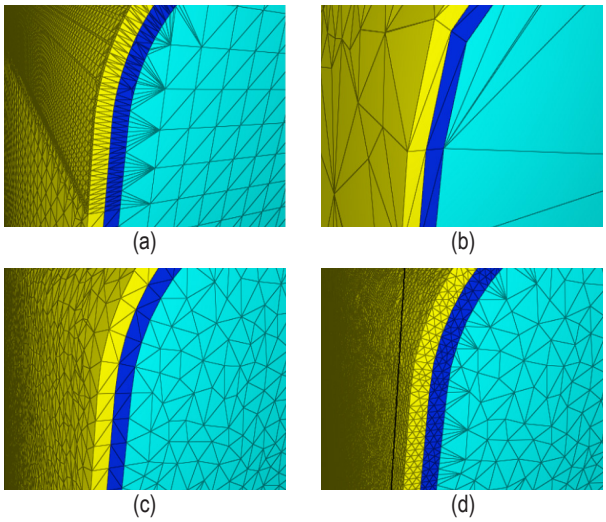


Figure 1-18: Closeup of tetrahedral elements on the top of the structure shown in Figure 17 as we progress through the remeshing steps. a) Default stress mesh b) REMESH DELAUNAY MANIFOLD c) REFINE d) REFINE CONSTRAINED.

StressOnly Mode of Victory Process

StressOnly mode is an operational mode of Victory Process in the sense of the well-known

- cell mode (large layout structure building) or
- process mode (complex physical simulations).

The key advantage of stressOnly mode is that the input structure for stress simulation is not coming from Victory Process but from another tools like Victory Mesh, with the advantage that the stress simulation mesh can be highly optimized manually. On top of this, even structures created by the solid modeling capabilities of Victory Mesh can be used for stress analysis.

In the 2024 baseline version of Victory Process, the input structure for stressOnly mode can be 2D as well as 3D. Since the only mesh used by Victory Process in stressOnly mode is the stress mesh, the memory footprint of such simulation is smaller than when embedding the stress simulation in the full process simulation flow. Moreover, EXPORT in this mode is fast because no re-meshing operation is needed within the EXPORT call.

The stressOnly operational mode is started from an existing meshed structure (.str file) via the statement

```
LOAD LOADSTRESSMESH NAME=<structure.str>
```

The file "structure.str" must contain the geometry and the mesh (triangular mesh or tetrahedral mesh) on which the stress analysis shall be performed and hence the input file must be a .str file. Both 2D and 3D structures are allowed. Typically, the input file is the output of a remesh step using Victory Mesh.

StressOnly mode supports only statements which are related to a `STRESS` statement, e.g.: `MATERIAL, METHOD, SETSTRESS, STRESSBOUNDARYCONDISIONS, EXPORT`. Note that the `SAVE` statement is not supported. This also means that only the `EXPORT` statement shall be used to create output when Victory Process is running in stressOnly mode.

1.6 The CMOS Diffusion Model Framework

The 2024 baseline release of Victory Process introduces a new diffusion model framework with the goal of addressing the out of the box predictive requirements for implantation and diffusion steps in CMOS manufacturing. The model framework is called the CMOS model in which unpaired active dopants will pair with defects to diffuse. Unpaired dopants are not mobile. This means that all effects due to out of equilibrium defects are taken into account with this model. The related equation system is described in the following.

The diffusion equation for the unpaired active dopant (X_L_A) is:

$$\frac{\partial}{\partial t} C_{x_A} = GR_{x_A} - GR_{xd,x\bar{d}}$$

Equation 10

The diffusion equation for the non-active dopant (X_Clu) is expressed as:

$$\frac{\partial}{\partial t} C_{x_{Clu}} = -GR_{x_A}$$

Equation 11

where GR_{xA} contains the reaction terms for the activation mechanisms.

The diffusion equation for the paired (active) dopant is written:

$$\frac{\partial}{\partial t} C_{xd,x\bar{d}} = \nabla\left(D_{xd,x\bar{d}}\left(\nabla C_{xd,x\bar{d}} + \rho_{xd,x\bar{d}} C_{xd,x\bar{d}} \frac{\nabla n}{n}\right)\right) + GR_{xd,x\bar{d}}$$

Equation 1-12

where ρ is the normalized charge (+1, -1,etc.) of the dopant and n is the free electron concentration.

The diffusion equation for the point defects is written:

$$\frac{\partial}{\partial t} C_{d,\bar{d}} = \nabla\left(D_{d,\bar{d}} C^{eq}_{d,\bar{d}} \nabla\left(\frac{C_{d,\bar{d}}}{C^{eq}_{d,\bar{d}}}\right)\right) - GR_{xd,x\bar{d}} - GR_{d\bar{d}}$$

Equation 1-13

where $G_{d\bar{d}}$ contains the reaction terms for the Frenkel pairs recombination as well as all other mechanisms af-fecting the point defects.

The pairing reactions rates are expressed as follows:

$$GR_{xd,x\bar{d}} = GR^{x+d\rightarrow xd}_{xd} - GR^{xd\rightarrow x+d}_{xd} + GR^{x\rightarrow xd+\bar{d}}_{xd} - GR^{xd+\bar{d}\rightarrow x}_{xd}$$

Equation 1-14

This model also includes some new additional submodels for defects and dopant defect pairs necessary to predict effects relevant in CMOS technology. Because the CMOS model takes into account the effect of out-of-equilibrium defects on diffusion, it is recommended to accurately describe the temperature ramps of the annealing processes in any simulation study using this model. These transient temperature phases are important sources of out of equilibrium diffusion and thus should be modelled for increased accuracy.

In the 2024 baseline version of Victory Process the CMOS diffusion model can be selected using the following METHOD statement:

```
METHOD MATERIAL="silicon" MODEL="cmos"
```

We offer the best possible predictability by means of a cali-bration file which is not part of the 2024 baseline package, but is provided to all users who express interest to work with the new CMOS diffusion model. This calibration file contains all the necessary statements for model flavor se-lection as well as the latest calibrated parameters for all diffusion-related parameters. It is widely recommended to use it when running the CMOS model. It is simply sourced in a deck with the following statement:

```
SOURCE <path_to_calibration_file>
```

Activation Models of the CMOS Model

The *CMOS* models uses the transient activation, for which there are three different flavors available:

i) Simple Transient Activation (STA) model

ii) Transient Activation & Deactivation Model (TAD) model

- Similar to STA model but allowing deactivation to happen

iii) Modified Suzuki (mSuzuki) model

- This more complex model involves the defect concentrations in the activation mechanism

When using the above-mentioned calibration file, the mSuzuki model is the default. The activation model can be selected using a `METHOD` statement

```
METHOD DIF.ACTIVATION.TRANSIENT="msuzuki"
```

Nucleation Clustering Dissolution (NCD) Model for Interstitials of the CMOS Model

Any implantation process generates crystal damage, represented in the context of the *CMOS* model as interstitials, vacancies and interstitial clusters. To obtain those, any implantation step must be simulated with the Monte-Carlo method (parameter `BCA` of `IMPLANT` statement) and the damage prediction must be activated in the implantation steps (parameter `PLUS.N` of `IMPLANT` statement).

The *CMOS* model introduces a new approach for the interstitials and interstitial clusters, called the NCD model. NCD stands for Nucleation Clustering Dissolution model as it models these three reaction mechanisms that can occur between interstitials and interstitial clusters. Typical values of the corresponding individual rates used in this model and the total rate are shown in Figure 1-19. This model is an extension of the 311-model and allows for exchanges between interstitials and interstitial cluster species to go both ways. It is enabled by default in the CMOS calibration file, but can be switched ON (or OFF) by means of a `METHOD` statement:

```
METHOD DIF.MODEL.NCD=ON
```

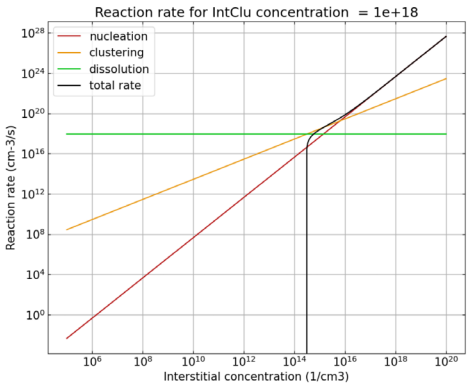


Figure 1-19 : Illustration of the evolution of the different mechanisms reaction rates and the total rate as a function of the interstitial concentration, for an interstitial clusters concentration of 10^{18} cm^{-3}.

Fluorine Models of the CMOS Model

In CMOS technology boron doping is often obtained though the implantation of the molecular species BF2 which allow for shallower implantation and also less diffusion afterwards. This effect on diffusion is particularly important for CMOS processing and such aspect the fluorine models are addressing.

There are two fluorine related models to choose from

i) the analytical model

ii) the more physical-based fluorineFDT3 model

In the analytical model the fluorine effect is taken care of by means of a correction factor applied to the dopant's diffusivity (only boron at the moment). The value of this correction factor (as it stands for the baseline 2024) is shown in Figure 1-20.

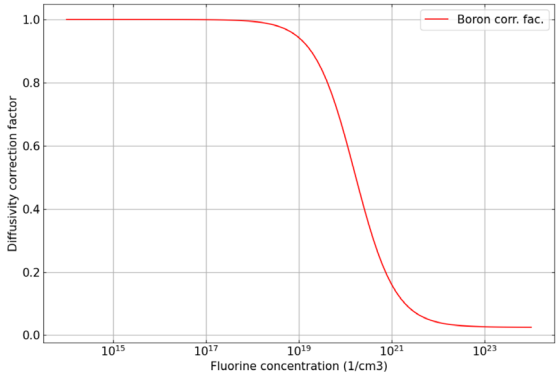


Figure 1-20: Evolution of the Boron diffusion correction factor as a function of the total Fluorine concentration (the values are related to the baseline 2024).

On the other hand, the fluorineFDT3 model adds three additional reactions to the CMOS model equation system,

i) the formation of fluorine-vacancy clusters,

ii) the trapping of interstitials by fluorine-vacancy clusters (or formation of fluorine-interstitial clusters)

iii) the dissolution of the fluorine-vacancy clusters.

In the CMOS calibration file, the default model is the analytical model, as it provides the best trade-off between simulation time and accuracy of results. Nevertheless, the fluorineFDT3 model can be enabled by means of the `METHOD` statement

```
METHOD DIF.MODEL.FLUORINEFDT3=ON
```

The fluorineFDT3 model can be enabled when the user wants to obtain very accurate and highly predictive results. Obviously, this comes with the price of longer simulation time, because a more complicated equation system is solved.

Arsenic-Phosphorus Co-diffusion Effect

The arsenic and phosphorus co-diffusion model is an analytical model very similar to the analytical fluorine model described above.

For phosphorus, it multiplies the diffusion of the phosphorus-defects pairs by a function of the arsenic concentration as illustrated in Figure 1-21.

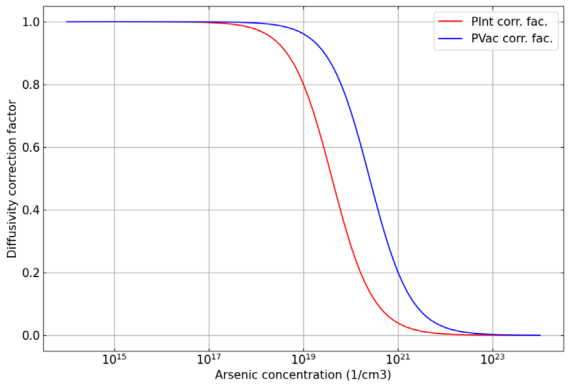


Figure 1-21: Phosphorus-defect pairs diffusivities correction factors as a function of the arsenic concentration at temperature of 1000°C (the values are related to the baseline 2024).

For arsenic, it multiplies the arsenic-defect pair diffusivities by a function of both the active arsenic and the active phosphorus concentration. To achieve this, the open material database interface of the 2024 baseline version of Victory Process introduces special function types. Those are called AsIntFactor and AsVacFactor. The hereby modelled diffusivity correction factor is shown in Figure 1-22.

In contrast to models modifying the equation system, these material database related models are enabled by means of the MATERIAL statement (not by means of the METHOD statement).

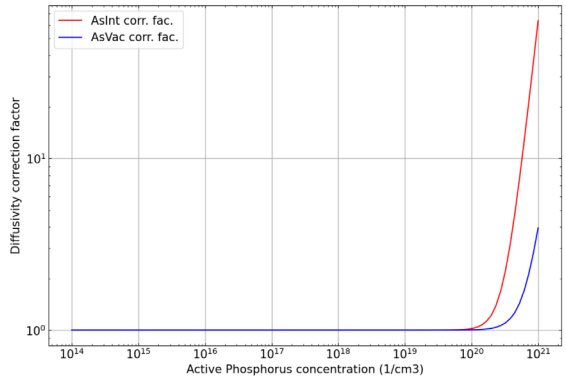


Figure 1-22: Arsenic-defect pair diffusivities correction factors as a function of the active phosphorus concentration for an active arsenic concentration of 1e18 and a temperature of 1000°C (the values are related to the baseline 2024).

1.7 Performance Improvements in the Diffusion Module

In the 2024 baseline version of Victory Process, we have considerably reduced the memory requirements and improved the speed of the diffusion module. This applies to DIFFUSE steps in inert as well as in oxidizing ambient when doping is present in the simulation flow.

Most of the performance gain could be obtained by code and software architecture optimization tailored to performance. This includes code reorganization, smarter algorithms, and unique numerical tricks. To summarize, the key changes are:

i) Unify stream classes, reduce number of operations, remove redundant data storages, and generalize members

ii) Store derivatives over charge in each point for all materials

iii) Assemble streams and reactions for interfaces in one go

iv) Improve assembling of interface equations for points with multiple materials

v) Adjust unphysical calibration parameters in diffusion equations, reaction equations, interface equations and equilibrium conditions, which introduce unnecessary stiffness to the equation systems

In the context of the numerical engine for diffusion also new and faster linear solvers are available. The default configurations of these linear solvers are further optimized, and the configurability of the linear solver settings is improved. Besides the default solver BICGSTAB, the new solvers PARDISO and HYP are available to solve diffusion and oxidation related equation systems, whereby especially PARDISO contributes to better simulation performance. Other supported linear solvers are AMS, PAS, SPD, XMS, and PAM. One can change the linear solver for the doping diffusion problem in the diffusion module with the statement:

```
METHOD DIF.SOLVER="pardiso"
```

In addition, many linear solver related settings can be modified like preconditioner, tolerance, max number of iterations, fill level, fill ratio, partitioning, ILU type:

```
ETHOD   DIF.PRECONDITIONER="ilk" \
        DIF.LINTOLERANCE=1e-6 \
        DIF.NBITERATIONS=2000 \
        DIF.LINFILLLEVEL=1 \
        DIF.LINFILLRATIO=1e-13 \
        DIF.LINPARTITIONING=1  \
        DIF.LINILUTYPE=0
```

Some of the linear solvers support multi-threading which can also be activated by means of the `METHOD` statement

```
METHOD DIF.SOLVERMULTITHREADING=ON
```

Since PARDISO supports multi-threading, it can perform better in some applications than the default linear solver BICGSTAB which does not support multi-threading, especially when the problems size is large.

Table 1-4 shows the performance of few diffusion test cases comparing baseline 2023 (version 8.18.1.R) vs. baseline 2024 (version 8.42.2.R) and BICGSTAB vs. PARDISO.

i) Example 1 features the FERMI diffusion model in a 2D structure with base phosphorus doping in silicon, then BF2 implant, deposition of oxide layer, that is implanted then with arsenic and indium, followed by long mid-temperature annealing.

ii) Example 2 features the TWODIM diffusion model in a 2D structure with silicon-oxide stack, followed by BF2 implant using plus.one model that introduces damage and hence point defects into simulation environment, which are then annealed with a rapid thermal step.

iii) Example 3 features CMOS diffusion model in a 3D structure with base BF2 implant into silicon, deposition of oxide, additional phosphorus implant, followed by ramp-up and finally ramp-down annealing.

Version	Solver	Ex. 1 - Fermi	Ex. 2 - Twodim	Ex. 3 - CMOS
8.18.1.R (2023)	BICGSTAB	42	154	458
8.42.2.R (2024)	BICGSTAB	34	136	96
	PARDISO, -P 1	33	125	147
	PARDISO, -P 4	23	81	89
*Values are in [minuntes]				

Table 1-4: Compare simulation time of VictoryProcess version 8.18.1.R (baseline 2023) and 8.42.0.R (baseline 2024) in 3 diffusion test examples. The recorded time refers to the simulation time in DIFFUSE step only.

Section 2: Device Simulation – New Features in 2024 Baseline Release

The following new features are now available in the 2024 Victory Device baseline release.

2-1 General

2-1-1 Input

- Support for multiple electrode names in the QFN.BIAS and QFP.BIAS parameters on the SOLVE statement, e.g.: solve qfn.bias="gate ngate".
- Support for saving the (absolute) trap energies to the structure file (TRAP.ENERGY on the OUTPUT statement).
- Support for Gaussian, complementary error function, and junction doping parameters on the DOPING statement (X.GAUSSIAN, Y.GAUSSIAN, Z.GAUSSIAN, X.ERFC, Y.ERFC, Z.ERFC, X.JDOPING, Y.JDOPING, Z.JDOPING, X.JUNCTION, Y.JUNCTION, and Z.JUNCTION).

2-1-2 Output

- Support for RESISTIVITY on the OUTPUT statement to specify that the resistivity will be saved to structure files.
- Support for probed quantity units in log file output.
- Support for traps and defects identification by a user-defined label, which can be used to control the output of trap-related quantities specified by OUTPUT and PROBE statements.
- Support for DIFFERENCE option on the PROBE statement to allow relative differences to the saved in log files.

2-1-3 Numerics

- Support for efficient, high-precision calculation of the error-function (erf), and complementary-error-function (erfc). This improves the precision of the trap-tunnel, Gaussian DOS, Anisotropic Mobility, and Single-Event-Upset (SEU) models, as well as certain DOPING calculations.
- Support for PAM and PAS linear solvers on Rocky Linux 9.

2-1-4 Error Handling

- Support for extra error message if the simulation exits due to non-convergence and FAIL.QUIT is specified on the METHOD statement.

2-2 Material Parameters

- Default values of C11, C12, and C44 for GaN/InN/AlN (and, by linear interpolation, for their ternary alloys and for InAlGaN) from Vurgaftman, I., et al., "Band parameters for III-V compound semiconductors and their alloys," Journal of Applied Physics, Vol. 89, No. 11 (2001): 5815-5875
- Default model for EP.MBULK
- Support for parameter FB.MBULK on the MATERIAL statement

2-3 Models

2-3-1 Contacts and Interfaces

- Fermi-level pinning model (Cowley and Moench formulations)
- Randomized and user set grain work-functions.
- Support for INT.RESIST parameter on the INTERFACE statement which sets an ohmic resistance between a conductor and semiconductor region
- C-Interpreter interface charge functions (F.CHARGE, F.QF, and F.QF.TIME on the INTERFACE statement)

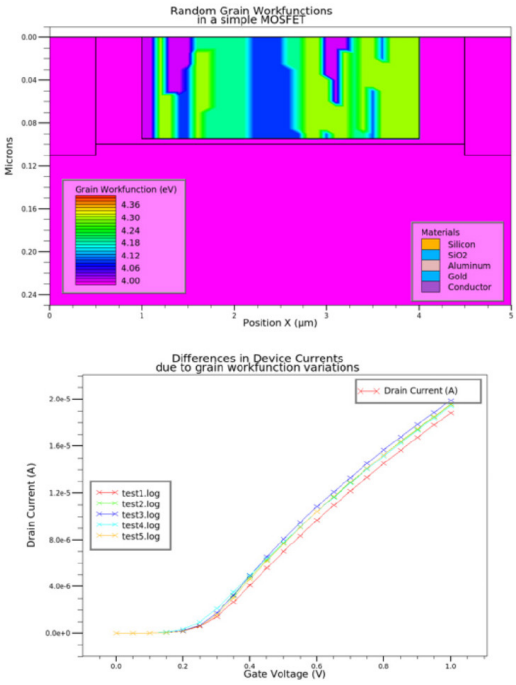


Figure 2-1: The effect of metal grain workfunction variations. The upper graph shows workfunction variations due a randomly generated pattern of metal grains. The lower part shows the effect of these on the IV characteristics of a simple MOSFET. The curves were generated by re-randomizing the grains for each voltage sweep.

2-3-2 Degradation

- Four-state non-radiative multi-phonon (NMP) model for NBTI and PBTI effects
- Four-state non-radiative multi-phonon (NMP) model This model allows to investigate device reliability with respect to NBTI and PBTI and can be enabled by set-ting the flag NMP4STATE on the DEGRADATION statement
  - o The four-state NMP model is dominated by the re-coverable component, which is ascribed to charge trapping into and out of bistable insulator defects from the adjacent semiconductor/insulator inter-face. The properties of those defects must be specified by electron/hole capture/emission processes are defined by the defect level (NMP4.ET1) and the corresponding relaxation energies (NMP4.S12S, NMP4.R12S) along with the parameters for the defect transformation (NMP4.EPST2s, NMP4.EPS2S2). The field dependence of the electron/hole emission times is also affected by the switching trap level (NMP4.ET2) and the corresponding relaxation energies (NMP4.S1S2, NMP4.R1S2) with the thermal barrier (NMP4.EPS1s1). The figure below demonstrates the asymmetry between stress and relaxation, which is an important feature of a reliability model. This asymmetry allows for long relaxation times (up to 100ks) compared to the short stress times (up to 1s).
  - o On top of the recoverable component, the four-state NMP model also accounts for the permanent component, which is represented by an interface reaction with the attempt frequency NMP4.NUP, the reaction energy NMP4.ED, the reaction barrier NMP4.EA, and the field dependence parameter NMP4.GAMMA.
  - o The model can also account for charge trapping to and from the gate (using the parameters NMP4.VG, NMP4.NG, NMP4.VG) and allows for the integration of trapping from all band states instead of only the band edges (setting the flags NMP4.BAND.INTEGRAL).

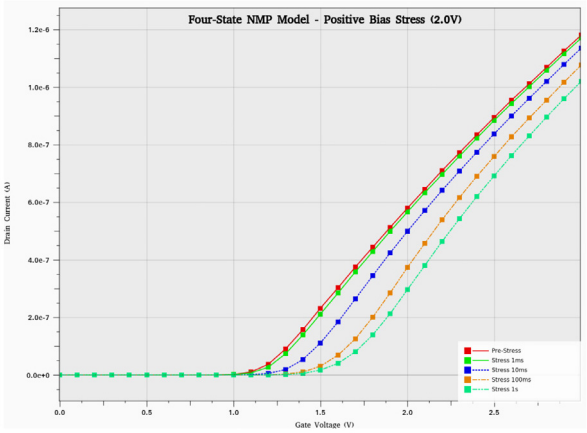


Figure 2-2: Device degradation (in a MOSFET) due to positive bias stress, associated with a shift of the transfer characteristics towards the right.

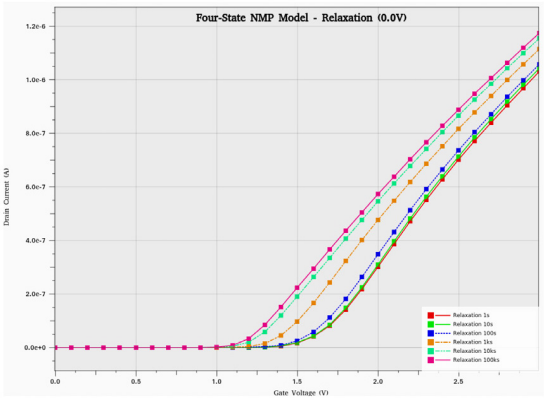


Figure 2-3: Device recovery after positive bias stress, visible as a shift of the transfer characteristics backwards towards the left.

The Figure 2-4 show the non-uniform distribution of trapped charges, due to the defect-to-defect variations considered by the four-state NMP model.

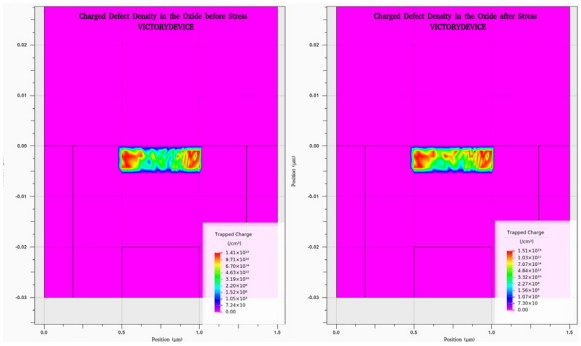


Figure 2-4: Trapped charge before (left) and after (right) positive bias stress.

2-3-3 Ferroelectric

- Support for new Ferroelectric polarization model based on Fedorova et al, Phys. Rev. B, 106,165122(2022).
- Support for new flag (LKFERRO.ITERATIVE on the METHOD statement) to specify an alternative way of solving the transient Landau-Khalatnikov equation in the LKFERRO model.

2-3-4 Impact Ionization

- NIDA Impact ionization model (4H SiC).
- Support for POST.PROCESS flag on the IMPACT statement which causes the generation rate due to impact ionization to only be calculated when a structure file is written from the SAVE statement without including the impact generation rate in the current continuity equations.

2-3-5 Incomplete Ionization

- Two-level cryogenic incomplete ionization model (INC.CRYO_TWO_LEVEL on the MODELS statement), which combines the cryogenic incomplete ionization model with two dopant levels.

2-3-6 Memory

- Support for direct tunneling coupling between the traps in the Silicon Nitride layer and the device channel in a SONOS memory device (SONOS.ERASE on the MODELS).
- Capability to charge up SONOS devices by using hot carriers injected from the channel into the gate stack. To use this model the gate stack insulators must be changed into wide bandgap semiconductors and the traps set up in the Silicon Nitride layer with the NITRIDECHARGE statement. The flag LUCKY.INJECT must also be set on the INTERFACE statement. On the MODEL statement both HEI and SONOS.HOT.EL must be set for electron charging, and HHI and SONOS.HOT.HO must be set for hole charging.
- Support for SIS.EL and SIS.HO models to simulate tunneling of electrons and holes respectively from one semiconductor region to another through an insulator region.
- Support for log file probe of Sonos Trap-to-Channel tunneling rate (E.T2CRATE on the PROBE statement for electron tunneling and H.T2CRATE on the PROBE statement for hole tunneling).
- Support steady state Ielmini Trap Assisted Tunneling models.
- Support for SIS.TAT option for the ITAT models. This al-lows trap-assisted-tunneling between two semiconductor regions, through an insulator barrier. Use with the RTAT.SC or ITAT.SC.EL or ITAT.SC.HO parameters on the MODELS statement for B2B, electron and hole tunneling respectively. Needs a Quantum Tunneling mesh.
- Support for the ability to specify traps for the SIS.TAT model to the DOPING statement, by specifying the TAT.TRAP flag.
- Support for the inclusion of TAT.TRAP charge in the Poisson equation. Can be disabled by clearing the TAT.POISSON flag on the MODELS statement.
- Support for thermal capture and emission terms to the SIS.TAT model for trap-assisted-tunneling through a barrier. These are vertical processes, just the same as for an ordinary trap, but are calculated on the quantum tunnel mesh. They are disabled by default but can be enabled with ITAT.THERMAL on the MODELS statement.
- Support for the ability to selectively enable/disable all 8 coupling components for the TAT.TRAPS in the SIS.TAT model. For the nonlocal tunneling models these are:[MODELS] ITAT.TUNNEL.ELEC.LEFT ITAT.TUNNEL.HOLE.LEFT ITAT.TUNNEL.ELEC.RIGHT ITAT.TUNNEL.HOLE.RIGHT. These are TRUE by default but can all be disabled using the ITAT.TUNNEL flag. For the thermal (local) terms the flags are [MODELS] ITAT.THERMAL.CAPT.E ITAT.THERMAL.EMIT.E ITAT.THERMAL.CAPT.H ITAT.THERMAL.EMIT.H. These are FALSE by default, all can be enabled with the ITAT.THERMAL flag.

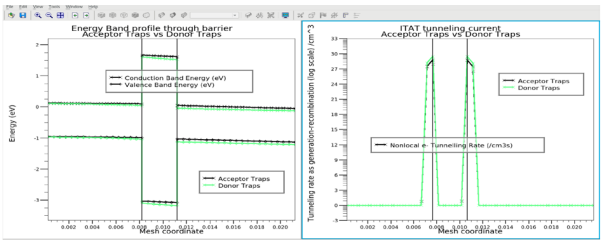


Figure 2-5: Tunneling current example for SIS.TAT ITAT.SC.EL model, showing the energy band line-up and the electron tunneling current at the interfaces of the barrier.

Figure 2-5 the energy band profile for a 3nm thick barrier of SiN between two N-type silicon regions in the left hand pane. In the right hand pane it shows the tunneling current produced by the SIS.TAT ITAT.SC.EL model. Results are shown for acceptor-type traps and donor-type traps in the SiN. The bias drop across the device is 0.5V.

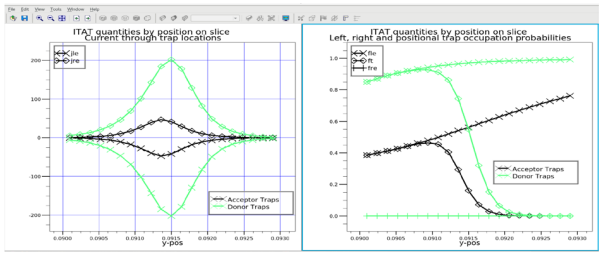


Figure 2-6 : Tunneling current example for SIS.TAT ITAT.SC.EL model, showing details of the tunneling current through the traps as a function of position.

Figure 2-6 details of tunneling along one slice through the 3nm SiN barrier. In the left hand pane the contribution to the overall current from each trap position is shown. The contributions to the left side and right side of the barrier are shown (source and sink), and tunneling current continuity is obtained. In the right hand pane, the occupation probabilities driving the current are shown. Fle corresponds to the occupation probability for the trap level at the left hand side Silicon, and Fre to the trap level for the right hand side silicon. Ft is the actual trap occupation probability. Fle-Ft and Ft-Fre drive the current.

Figure 2-7 shows the current as a function of bias through the tunnel barrier structure. The difference between the acceptor trap case and the donor trap case is due to electrostatics causing a shift in the band line-up. The case with the model not enabled is also shown.

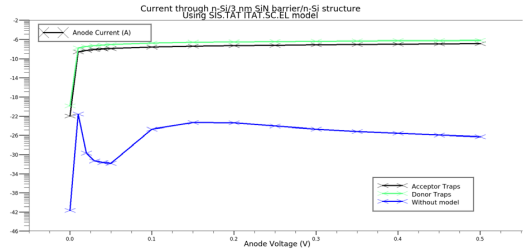


Figure 2-7: Tunneling current example for SIS.TAT ITAT.SC.EL model, showing the terminal current as a function of bias.

Some relevant deck commands are

trap region=2 tat.trap density=5.0e18 donor sign=1.0e-12 sigp=1.0e-12 e.level=3.05 degen.fac=2

models srh fermi ni.fermi itat.sc.el sis.tat qtregion=1

- Support for thermal capture and emission terms to the SIS.TAT model for trap-assisted-tunneling through a barrier. These are vertical processes, just the same as for an ordinary trap, but are calculated on the quantum tunnel mesh. They are disabled by default but can be enabled with ITAT.THERMAL on the MODELS statement.
- Support for the ability to selectively enable/disable all 8 coupling components for the TAT.TRAPS in the SIS.TAT model. For the nonlocal tunneling models these are:[MODELS] ITAT.TUNNEL.ELEC.LEFT ITAT.TUNNEL.HOLE.LEFT ITAT.TUNNEL.ELEC.RIGHT ITAT.TUNNEL.HOLE.RIGHT. These are TRUE by default but can all be disabled using the ITAT.TUNNEL flag. For the thermal (local) terms the flags are [MODELS] ITAT.THERMAL.CAPT.E ITAT.THERMAL.EMIT.E ITAT.THERMAL.CAPT.H ITAT.THERMAL.EMIT.H. These are FALSE by default, all can be enabled with the ITAT.THERMAL flag.

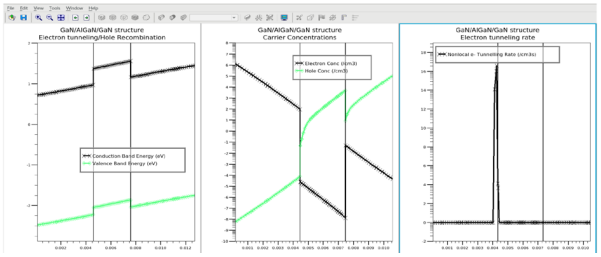


Figure 2-8: Tunneling current example for SIS.TAT ITAT.SC.EL model. Electron tunneling from one side of the barrier is compensated by local holecapture from the quantum well. Flags ITAT.THERMAL and ITAT.TUNNEL.ELEC.LEFT are activated.

Figure 2-8 shows an example of non-local electron tunneling into traps in a quantum barrier being coupled with local hole capture from the valence band. The left hand pane shows the energy band profile through the model GaN/AlGaN/GaN structure, and the carrier densities. The central pane shows the carrier densities and the right hand pane shows electron tunneling rate from the GaN at the left hand side.

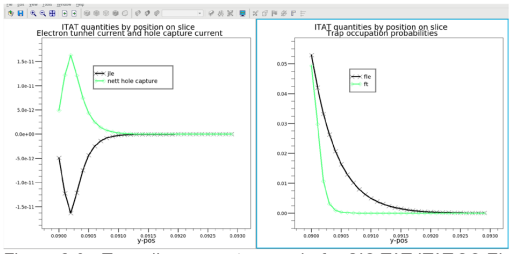


Figure 2-9 : Tunneling current example for SIS.TAT ITAT.SC.EL model. Details of the electron tunneling and hole capture versus position for the results shown in figure 2-8.

Figure 2-9 shows a typical slice through the AlGaN barrier on which the tunnel and recombination current is calculated. In the left hand side pane, the contribution to the tunnel current from each trap position is shown, along with the hole recombination rate. The rates balance each other locally and so give good current continuity. The right hand pane shows the occupation probability of electrons in the GaN at the trap level(Fle), and the actual occupation probability of the trap (Ft). The tunnel current is driven by Fle-Ft. The hole capture rate is driven by (1-Ft) because Ft is an electron occupation probability. The relevant models specification is

- models srh fermi ni.fermi itat.sc.el sis.tat qtregion=1 ^itat.tunnel.elec.right itat.thermal.capt.h
- Here itat.sc.el enables the tunneling model for electrons ^itat.tunnel.elec.right turns off tunneling from the right of the device and itat.thermal.capt.h enables hole capture from the valence band.

2-3-7 Mobility

- New tabular stress-mobility models. This al-lows the relative stress directions (e.g. MOBIL-ITY STRESS.LONG="1,1,0" STRESS.VERT="0,0,1" STRESS.TRANS="-1,1,0") along with tabular data files of relative mobility vs stress in these directions (e.g. MOBILITY ORTHO.N.LONG="n-long.dat" ORTHO.N.TRANS="n-trans.dat" ORTHO.N.VERT="n-vert.dat"), to recreate almost any stress-mobility relationship. This model can be used with surface orientation.
- Surface orientable scaling factor for the electron and hole bulk mobilities.
- Prefactor for the electron mobility of the host semiconductor in the bridging mobility model. This field-, temperature-, and species concentration-dependent prefactor can be defined by using a C-Interpreter function and is enabled by setting F.BRFACTOR on the MOBILITY statement.
- Support for the parameters CVT.N.SFACTOR and CVT.P.SFACTOR to the MOBILITY statement. These set the fractional stress mobility dependence (from the bulk mobility model) which will be applied to the non-bulk cvt terms.
- Support for the VSAT.QF (VSAT.QFN, VSAT.QFP) and VSAT.ZAK (VSAT.ZAKN, VSAT.ZAKP) options to the MOBILITY statement. These force the fldmob and gansat models to use the gradient of the quasi-Fermi level or sqrt(E * grad_qfl) instead of the electric field as the driving force.
- Support for the F.MUVSATN and F.MUVSATP c-interpreter functions for saturation velocity.

2-3-8 Organic

- Distribution dependent singlet dissociation model.
- Support for gensinglet and gentriplet c-interpreter functions (F.GENSINGLET and F.GENTRIPLET on the MODELS statement).
- Support for PROFILE.GAUSS on the MATERIAL statement, to display the Gaussian band-structure.
- Support for LANGEVIN.FIELD.WEIGHT and LANGEVIN.MU.BULK parameters to the MODEL command. These control the mobility used in the Langevin recombination.

2-3-9 Polarization

- Support for angled crystal growth strain/polarization model for Wurtzite crystals grown at an angle to the substrate crystal axis. This is activated by setting EPI.ANGLE on the MATERIAL statement, along with CALC.STRAIN and POLARIZE on the MODELS statement.

2-3-10 Recombination

- Support for a new carrier and doping concentration dependent model for the Auger rate coefficients suitable for modeling silicon under high and low injection conditions. The model is enabled by specifying PIC.AUG on the MODEL statement.
- Support for new Auger recombination models which are valid under both low and high injection conditions. The Kerr model is enabled by specifying KERR.AUG on the MODELS statement and the Richter model is enabled by specifying RICHTER.AUG on the MODELS statement.
- Support for a new Auger recombination model and a new optical recombination model suitable for modelling Silicon solar cells. To enable the Auger term specify SOLAR.AUG on the MODELS statement. To enable the optical term specify SOLAR.OPTR on the MODELS statement.

2-3-11 Quantum

- Support for superlattice model to solve the 1D open-boundary Schrodinger equation, based on the quantum transmitting boundary method (QTBM).

  A new SLATT model has been added to Victory Device implementing the QTBM algorithm described in Chenjing L., and William R. Frensley. "An efficient method for the numerical evaluation of resonant states", Journal of applied physics, Vol. 76, No. 5 (1994): 2881-2886. It is a nonlinear eigenvalue solver which uses Newton's method to refine the eigenvalues obtained after an initial eigenvalue calculation. Nonlinearity is introduced by the open boundary conditions. The model is based on an effective mass description of the electron band structure. The model is enabled by specifying SLATT on the REGION or MODELS statements. Alternatively, the new SLATT statement can be used.

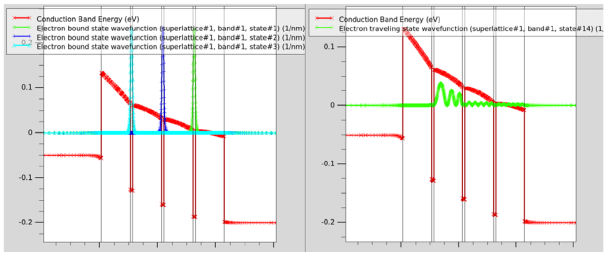


Figure 2-10. Bound state wavefunctions in each of the three GaAs QWs (left) and lowest traveling state wavefunction above the top of the leftmost QW (right).

The model calculates the resonant states in superlattices, including both bound and quasi-bound (or traveling) states. The latter are typically found in Quantum Well Infrared Photodetectors (QWIPs) at energies near the top of each QW barrier. The energies and wavefunctions of resonant states are saved to the structure file. Figure 2-10 shows the spatial profile of a few wavefunctions calculated for a photodetector consisting of three GaAs QWs sandwiched between Al0.25GaAs0.75 layers, which act as barriers for the confined carriers. .

- Support for HBAND6 model which calculates stress-induced hole valley shifts and effective masses for use by the MVSM model.
- Support for the QTREGION statement to allow the set up any number of quantum tunneling regions, each with its own set of quantum tunneling slices.
- Support for optical transitions between neighboring wells to the QWELL model. They are enabled by specifying WELL.WWSPONT on the MODELS statement together with WELL.CAPT and SPONTANEOUS for interband transitions or WELL.CAPT and INTERSUB.SPONT for intersubband transitions.
- Support for optical output coupling (optical efficiency) to the log files when led analysis is performed.
- Support for stress-strain conversion for Wurtzite materials. It uses the elastic stiffness constants specified by the MATERIAL parameters C11, C12, C13, C33, and C44.
- Support for log file probe of carrier recombination/generation rates caused by the BBT.NONLOC mode (R.NLEBBT on the PROBE statement for electrons and R.NLHBBT on the PROBE statement for holes).
- Support for output of intersubband spontaneous emission rates of electrons and holes (INTERSUB.SPONT model) to structure and probe log files. The probes are enabled by N.INTERSUB.SPONT and P.INTERSUB.SPONT on the PROBE statement.
- Support for the TAT.NONLOCAL model. It requires a quantum tunneling mesh. The trap level is taken as the intrinsic energy level, unless the TAT.NLDEPTH is specified on the MODELS statement. This is relative to the conduction band energy unless the TAT.RELEI flag is specified on the MODELS statement, in which case TAT.NLDEPTH is relative to the intrinsic energy. The tunneling masses are set with ME.TUNNEL and MH.TUNNEL parameters on the MATERIAL statement.
- Enabled SNRM.BEG, SNRM.END, STNL.BEG, STNL.END parameters of QTREGION statement.
- Support for optical transitions between neighboring wells to the QWELL model. They are enabled by specifying WELL.WWSPONT on the MODELS statement together with WELL.CAPT and SPONTANEOUS for interband transitions or WELL.CAPT and INTERSUB.SPONT for intersubband transitions.

The SPONTANEOUS and INTERSUB.SPONT models for light absorption and emission in QWs have been expanded to include transitions between states in neighboring QWs. The expressions for the optical transition rates are calculated by taking as inputs the energies and wavefunctions of bound states calculated by the QWELL model for the separate wells. The overlap integral between wavefunctions belonging to two different wells is calculated by merging the separate spatial grids of each well.

Figure 2-11 shows simulation results for an InAs/GaSb type-II SL photodetector.

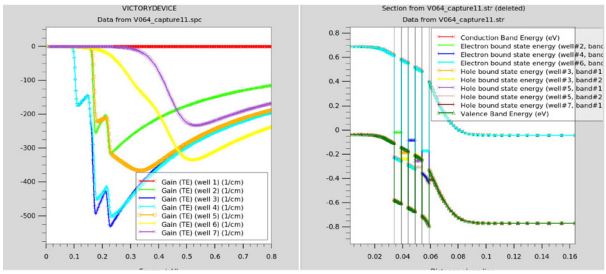


Figure 2-11. Spectrum of gain (negative of absorption coefficient) (left) and band diagram (right) for a type-II SL.

2-4 MixedMode

- Support for LENGTH on the MixedMode Txxx element. This specifies transmission line length in m.
- Support for MixedMode parser expressions on the Exxx statement. If VALUE or VOL is specified, the following statement format is used: Exxx node1 node2 VALUE | VOL=<expression>
- Support for MixedMode parser expressions on the Gxxx statement. If VALUE or CUR is specified, the following statement format is used: Gxxx node1 node2 VALUE | CUR=<expression>
- Support for DER.V(node) and DER.V(node1,node2) in the MixedMode expression parser. DER.V(node) calculates the derivative of V(node) with respect to time and DER.V(node1,node2) calculates the derivative of V(node1) - V(node2) with respect to time.

- Support for DDT() in the MixedMode expression parser. This calculates the derivative with respect to time. Currently this is limited to nodal voltages only (e.g. DDT(V(node1,node2)), i.e. equivalent to DER.V).
- Support for expressions in MixedMode PWL waveforms.
- Support for MixedMode function definitions via the .DEFINE or .FUNC statements.
- Support for PRINT.EXPRESSIONS on the MixedMode .OPTIONS statement. If this is specified, then element expressions will be printed every time they are updated.

2-5 Optics

- Support for photon absorption rate output for semiconductors, insulators and conductors for the raytrace and FDTD methods.
- Support for an Atlas matching option for reverse raytracing (RRT.ATLAS) to the LED and SAVE statements.
- Support for contribution of intersubband spontaneous emission to the log file output of "Radiative rate" and "Luminous power" enabled by the LED parameter on the REGION statement.
- Support for the inclusion of intersubband spontaneous emission (INTERSUB.SPONT model) into the power of optical sources in LED simulations.
- Support for Monte Carlo raytracing (MONTE.CARLO on the BEAM statement).
- Support for automatic optical simulation between two bias steps when ABS is set in the REGION statement, in agreement with Atlas.
- Support for slicing in the TMM solver. Slice width is uniform and can be automatic (up to a 1-nm precision) or user-defined (SLICE.WIDTH parameter in the BEAM statement).
- Support for Photon Detection Probability log file probe, which is used in conjunction with the BEAM statement and the GEIGER model. Specify PROBE PDP.PROBABILITY INTEGRATE to use. The domain of integration can be limited in the usual way if required. Works only for monochromatic beams at present. The GEIGER flag should also be specified on the MODELS or IMPACT statement in order to use this model.

Section 3: Victory DoE – New Features in 2024 Baseline Release

2024 baseline release of Victory DoE includes the following key features:

- Simulation deck(s) in Victory DoE are compatible with DeckBuild
- Users can take advantage of the flexibility of DeckBuild combined with the DOE capabilities of Victory DoE
- Simulation projects are UI driven
- Intuitive user interface enables easy navigation between projects
- Various sorting methods of simulation projects are provided such as sorting by flag, time, comment, file size, etc., helping to organize multiple projects
- Single Victory DoE tool manages multiple projects at the same time
- Simulation results are stored in Silvaco proprietary format as well as csv format
- Full integration with Victory Visual– Silvaco's TCAD results visualization tool
- Supports real time structure visualization function during the simulation
- Built-in data visualization tool (Chart Designer) for reviewing DOE results
- Supports LSF (Load Sharing Facility)

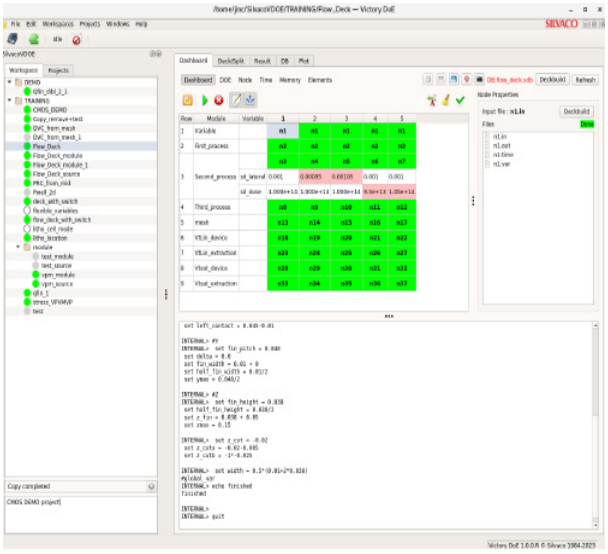


Figure 3-1.

Victory DoE is a UI driven software solution to automate TCAD simulation projects, run experiments and perform data analysis. It is designed to be compatible with Silvaco's DeckBuild environment, in which users have full control of simulation decks editing, debugging and visualization. Victory DoE also delivers cutting-edge DoE algorithms that boost simulation effectiveness.

3-1 Dashboard View

- Controls simulation progress, structure visualization, debug, and add / delete simulation conditions
- Monitors the simulation progress by reviewing output and structure files
- Displays simulation time for all nodes. Displays maximum memory usage and maximum number of nodes during simulation
- Enables addition and deletion of simulation conditions
- Modification of the "golden-deck" and re-run of the project

3-2 Simulation Project Workspace

- Manages simulation projects, which are organized like directory structures
- Supports multiple workspaces
- For each project, users can assign a flag and attach comments
- All essential file management tools are provided: copy (clean), paste, archive, clean project, lock, etc.
- Allows users to perform search and sort based on the file name, time, flag, file size, …etc.
- Shared workspace enables a collaborative workflow in which multiple users share projects
- Simulation examples are available through default example workspaces

3-3 Golden-deck and DOE Table

- Deck / Split View to construct a simulation project by assigning the Golden-deck with as supplemental file
- Define DOE input variables and create DOE tables
- Drop-zone for managing the deck and supplemental files
- Any set commands in the Golden-deck can be DOE in-puts
- Simulation version and number of threads are automatically assigned as split variables
- Users have the option to lock the simulation version per project
- DOE tables can be created manually or automatically using the DOE builder

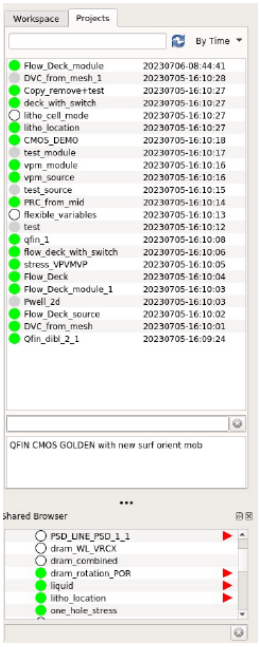


Figure 3-2. Project Workspace

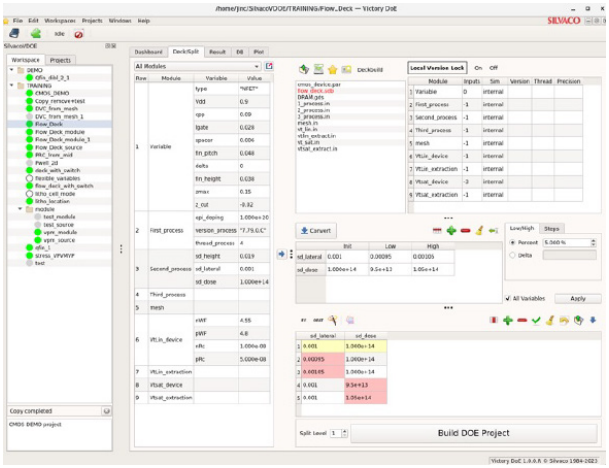


Figure 3-3. Deck/Split View.

3-4 Simulation Result View

- Victory DoE automatically generates the output value at every extract command from the golden-deck
- Input variables and simulation a available in a single table
- Download to .csv format is available

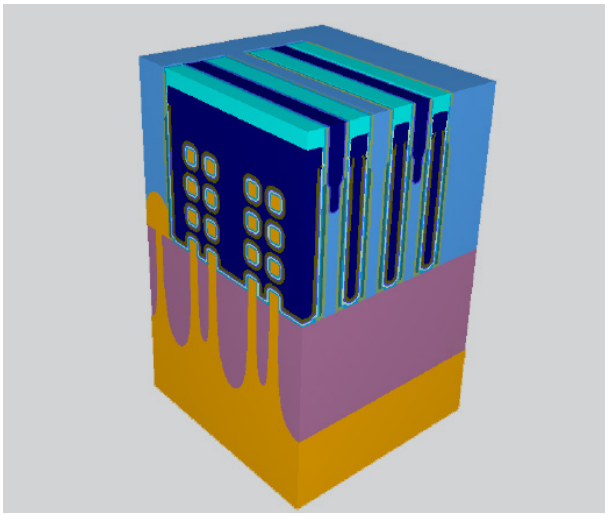


Figure 3-4. Simulation Structure Visualizationby Victory Visual.

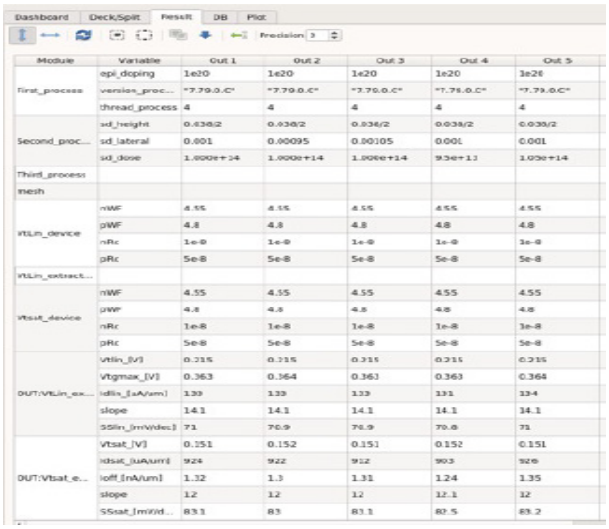


Figure 3-5. Simulation Results View.

Figure 3-5 Simulation Results View

- Integration of Victory Visual to view results
- All files in one location, allowing easy comparison of split lots
- Plot overlay between various conditions allows for quick analysis of different splits

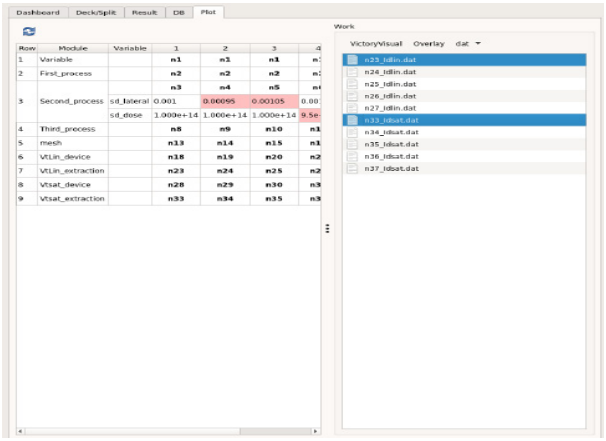


Figure 3-6: Plot View.

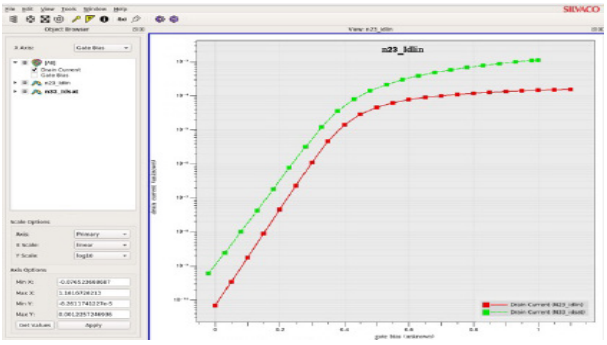


Figure 3-7: Victory Visual Overlap Plot.

3-6 Special Project Flow

1. Skeleton with multiple modules

- Inspired by the conventional software development flow in which a project is comprised of multiple modules. Each module can run independently so that it can be plugged into other projects
- The goal of the skeleton-deck is to build a deck using modules
- Splits are performed on modules
- Modules can be processing recipes or device characterization (IVs, CVs)
- Ideal for collaborative work environment (no requirement of "deep" TCAD simulation knowledge)

2. Switch function with superset golden-deck

- This flow is coming from the requirement that engineers want to have a single superset golden-deck and use only a portion of the flow. Examples may include a conventional CMOS project having NFET, PFET, Vtlin, Vtsat, breakdown, leakage, CV curves, etc. This method is advantageous in keeping the golden-deck golden
- Victory DoE offers a switch function with which users run a portion of the golden-deck more efficiently

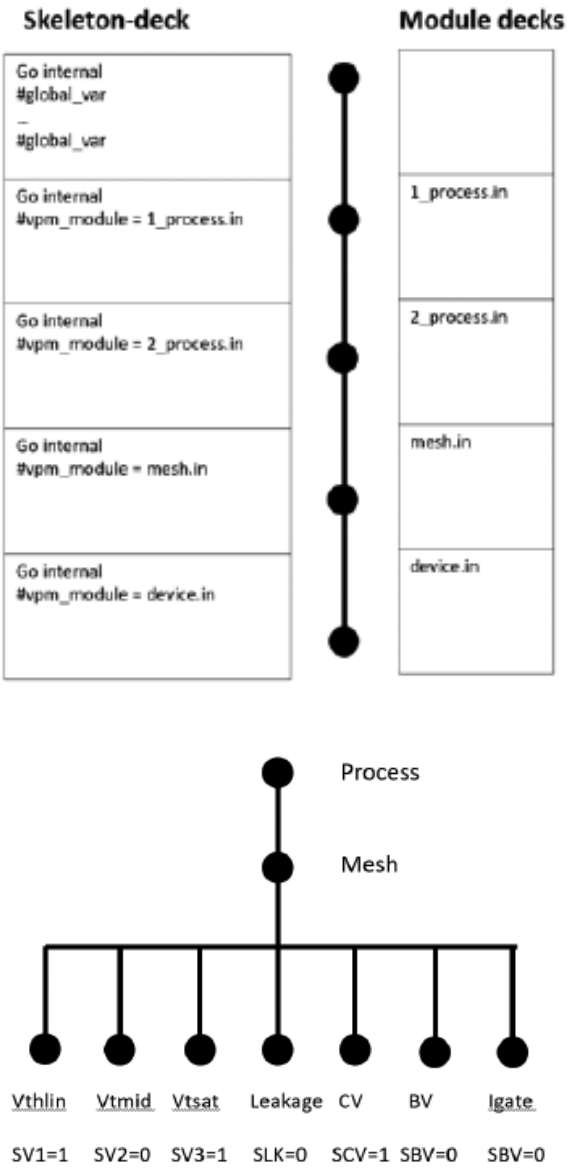


Figure 3.8: Schematics of Special Simulation Flow (A) Skeleton/ module Flow. (B) Switch Feature on the Single Golden-deck Flow.

3-7 Simulation Version Control

- Victory DoE provides straightforward management of simulator versions
- All Silvaco simulation tool versions are displayed in a simple table format
- Ability to lock the simulation tool version for each project

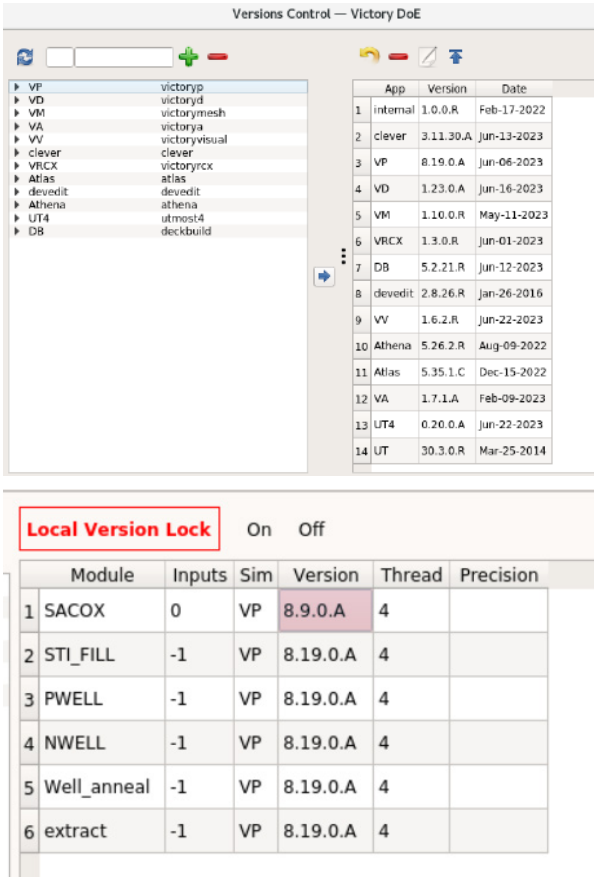


Figure 3-9: Simulation Tool Version Control (A) Global Version Control (B) Local Version Control.

3-8. DOE Builder

- Generate DOE table from various DOE algorithms
- Option to add Augmented and Constraints DOE features
- Provides state-of-the-art DOE algorithms such as computer-generated DOE from D-optimization algorithm
- Visualization of DOE table
- DOE builder supports Fraction of Design Space (FDS) plot
- Power analysis of DOE is conducted in the DOE builder

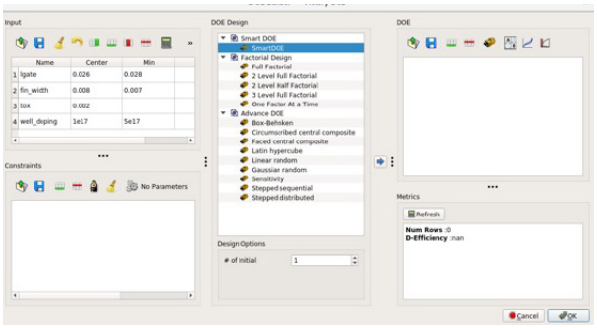


Figure 3-10: DOE Builder.

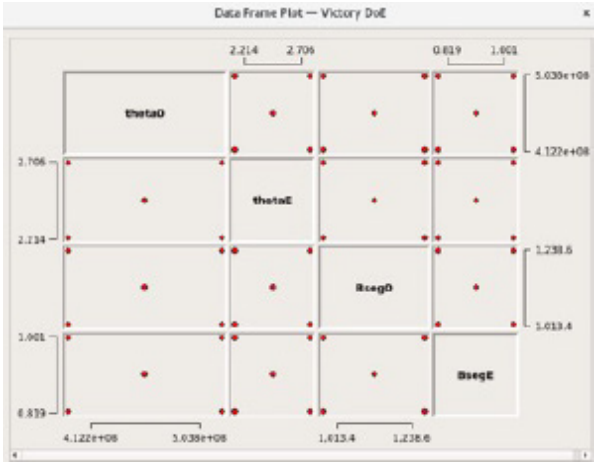


Figure 3-11: (A) DOE plot.

3-9. Chart Designer

- Chart Designer is a built-in data visualization tool that offers insight into the dataset
- X-Y plot with X-axis as a string value
- Bar chart
- Scatter plot
- Scatter plots Matrix

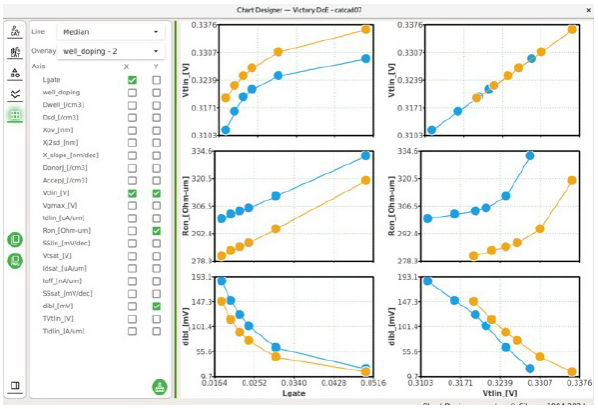


Figure 3-12

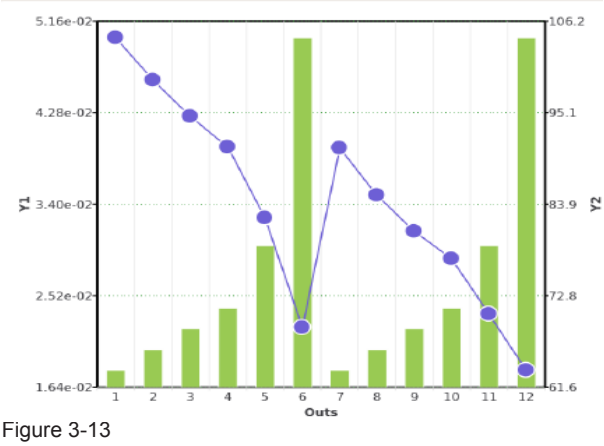


Figure 3-13