SYNOPSYS®

Sentaurus Technology Template: CMOS Processing

Abstract

This TCAD Sentaurus™ simulation project provides a template setup for deep-submicron CMOS process and device simulations for a 32-nm technology node.

This project setup is used to simulate a 32-nm CMOS process flow with the *gate-first scheme* and is intended to serve as a convenient starting point for any deep-submicron CMOS process simulation. The process flow includes advanced process steps such as shallow trench isolation, high-k dielectrics with metal gate to reduce leakage, laser annealing to suppress transient-enhanced diffusion, stress engineering with epi-SiC and epi-SiGe pockets, silicidation, and dual stress liner to improve the on-state drive current. Sentaurus Process uses the Advanced Calibration set of process models and parameters. The electrical characteristics of the CMOS devices are simulated using Advanced Calibration models and parameters in Sentaurus Device. The Sentaurus Workbench template project performs certain device simulations to extract key electrical parameters to facilitate customized calibration and optimization of projects.

Introduction

This project provides a standard template for performing deep-submicron CMOS process simulations. The setup of the Sentaurus Workbench project supports NMOS and PMOS devices, and allows for the simulation of structures with different gate lengths. Other variations can be included by adjusting the template. The process simulations enable the modeling of device geometry and dopant distribution in deep-submicron semiconductor technology using Sentaurus Process.

The simulated structure is used in device simulations to determine the electrical characteristics using Sentaurus Device. After the process simulation, the following device simulations are performed: I_d–V_{gs} curves for a low-drain bias and high-drain bias, a family of I_d–V_{ds} curves for various gate biases, as well as small-signal simulations that are performed to obtain the C–V characteristics of the respective devices. For each of the simulated curves, relevant electrical parameters are extracted.

The focus of this project is to provide a setup that can be used as is or customized easily according to specific needs for advanced use. The documentation focuses on tool-specific setups as well as the technology aspects of 32-nm CMOS process simulation. It is assumed that users are familiar with the Sentaurus tool suite, in particular, Sentaurus Workbench, Sentaurus Process, Sentaurus Device, and Sentaurus Visual as well as IC WorkBench Edit/View Plus (ICWBEV Plus). For an introduction and tutorials, refer to the TCAD Sentaurus Tutorial.

General Simulation Setup

This section describes the tool flow of the Sentaurus Workbench project. For each tool, the associated Sentaurus Workbench input parameters and the extracted parameters are discussed.

Sentaurus Process

The tool sequence for this project starts with Sentaurus Process, which generates the CMOS devices. The following Sentaurus Workbench parameters are used by Sentaurus Process:

- `CUT = SIM2N|SIM2P` defines which layout cutline is used to translate the 2D masks into 1D masks for use in Sentaurus Process. The cutline names `SIM2N` and `SIM2P` correspond to the NMOS and PMOS transistors, respectively.
- `Layout = gds|icwb` defines the loading of the layout file in Sentaurus Process. The `gds` option is used in this setup.
- `lgate` [µm] determines the gate length. It is used to stretch the layout and parameterize the grid definition. The values used in this project are 0.02 µm and 0.03 µm. Users can add or remove values as required.
- `f = 1|2` is the parameter used to scale the mesh density used during process simulation. A factor of 1 produces accurate results at a slower speed. A factor of 2 still gives reasonable accuracy, but runs considerably faster. Here, a factor of 1 is used.
- `ver = 1` is an optional parameter, which can be used to distinguish between different model parameters. The parameter is provided for easy customization only and is not used in the given setup.
- The subsequent parameters `Poly`, `Halo`, `Ext`, `SD`, `Backend`, and `Mesh` are logical parameters, which are used to split the process flow into separate groups of simulation steps that are executed individually. At the end of each group, the current state of the simulated structure as well as the current Sentaurus Process settings are saved and reloaded at the beginning of the next group. This approach allows easy access to the intermediate structures for verification and allows for split variations. For example, simulations for different extension implant conditions can be performed without having to re-simulate the processing steps before the extension implants. (See Sentaurus Process on page 5 for details about the different simulation splits.)

Sentaurus Process extracts the following parameters:

- `Lgeff` [µm]: Effective channel length defined as the distance from the source-side to the drain-side p-n junctions near the silicon–oxide interface.
- `Xj` [µm]: Source/drain junction depth.

Sentaurus Device and Sentaurus Visual

Sentaurus Device simulates I_d–V_{gs} curves for a low-drain bias and high-drain bias, a family of I_d–V_{ds} curves for various gate biases, and a small-signal simulation to obtain the C–V characteristics of the NMOS and PMOS devices.

For each simulated curve, relevant electrical parameters such as threshold voltages, drain current levels, and parasitic parameters are extracted with Sentaurus Visual. The device simulation and extraction setup is identical to the one in the Sentaurus Workbench template project *Sentaurus Technology Template: CMOS Characterization*. Refer to the documentation for that project for details about the device simulations, visualization, and parameter extraction.

Tool-specific Setups

IC WorkBench Edit/View Plus Interface

TCAD Sentaurus features an interface to the Synopsys layout editor IC WorkBench Edit/View Plus (ICWBEV Plus), which allows you to view and edit of layouts in most of the industry-standard formats including GDSII and OASIS. For details about this interface, refer to the *Sentaurus™ Process User Guide*.

There are two options for loading a layout file into Sentaurus Process:

- Option A: Load the GDSII file directly into Sentaurus Process.
- Option B: Create a TCAD layout file using ICWBEV Plus.

Option B allows you to assign interactively meaningful names to layers and to place simulation domains and stretches using the ICWBEV Plus interface. Option A allows you to perform the same tasks directly in Sentaurus Process using script commands.

NOTE Option A does not require access to ICWBEV Plus. However, you will need access to a GDSII viewer to determine the proper coordinates of the simulation domains to be used in the Sentaurus Process input file.

The layout interface in Sentaurus Process is initialized in either of the following ways:

- Option A: To read a GDSII layout file:

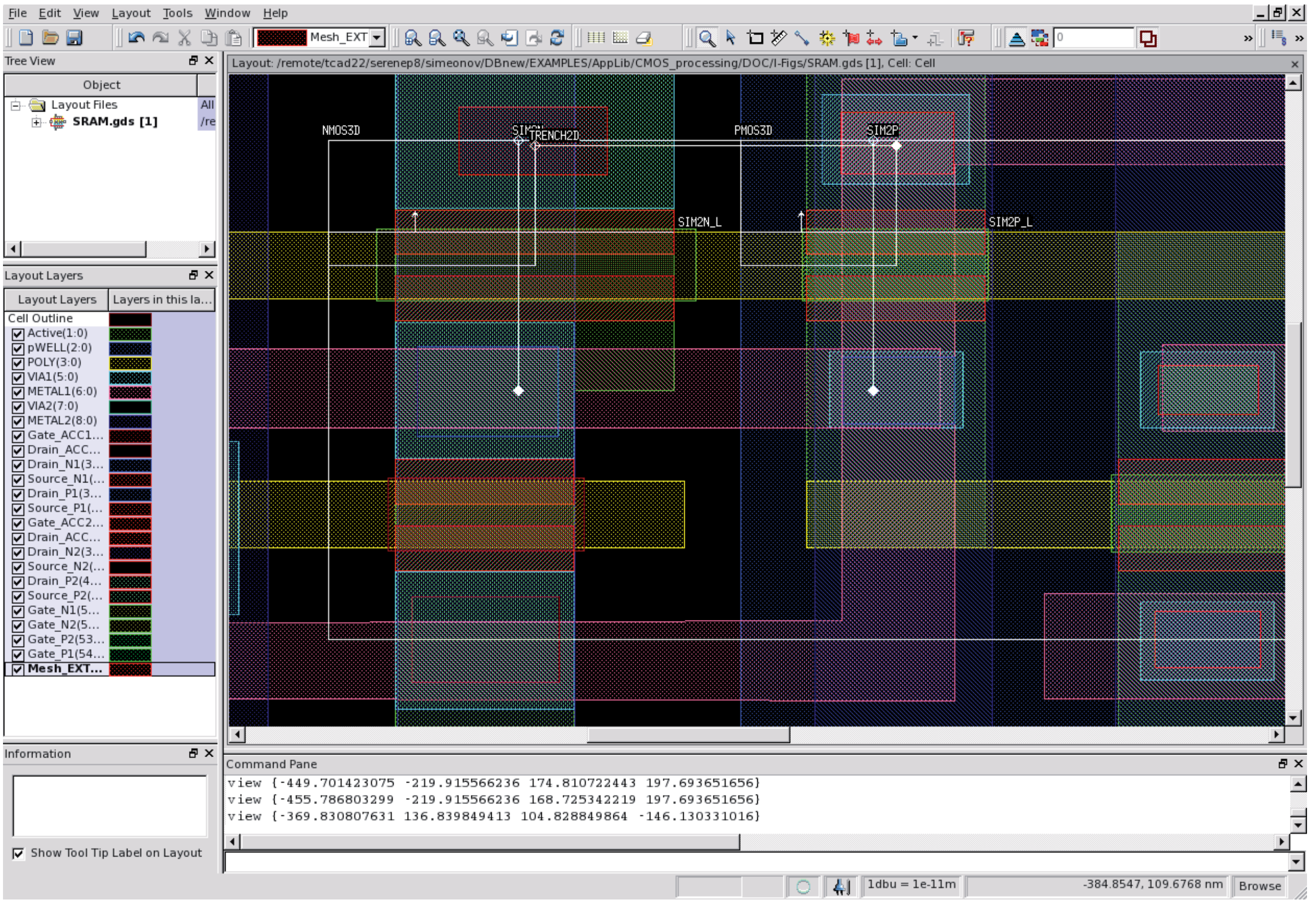
```
icwb gds.file= <GDSII filename> [scale=<scale>] ...
```

- Option B: To read a TCAD layout file:

```
icwb filename= "<filename.mac>" [scale=<scale>]
```

Option A requires that you specify some additional options such as layer properties, the stretch option, and the simulation domain in the GDSII coordinate system.

Figure 1 shows the layout together with the markups. In addition to the different layers (shaded polygons), there are several other objects specific to the current project simulation. The white lines marked `SIM2N` and `SIM2P` represent the 2D simulation domains of the corresponding transistors.



```
view {-449.701423075 -219.915566236 174.810722443 197.693651656}
view {-455.786803299 -219.915566236 168.725342219 197.693651656}
view {-369.830807631 136.839849413 104.828849864 -146.130331016}
```

Figure 1 ICWBEV Plus layout with the markups. The two vertical white lines (SIM2N and SIM2P) show the cutlines that define the 2D areas for the NMOS and PMOS devices. The layout is stretched as a function of the parameter lgate.

The markup file is used as a source of information to generate the TCAD layout file SRAM_lyt.mac, which contains a truncated version of the original layout with all the markup details. For option B, this file is used then as an interface to the process simulation:

```
icwb filename= "SRAM_lyt.mac" scale= 1e-3
```

For option A, you can load a GDSII file directly with specific layer properties using:

```
icwb gds.file= "SRAM.gds" cell= "Cell" scale= 1e-3 \
  layer.numbers= {1:0 2:0 3:0 5:0 6:0 7:0 8:0 ...} \
  layer.names= {Active pWELL POLY VIA1 METAL1 VIA2 \
    METAL2 ...} \
  sim2d= "-240.0 112.5 -240.0 0.0" \
  stretches= {MOS_L= {-325.0 71.25 -170.0 71.25}}
```

The Sentaurus Workbench variable @Layout@, which accepts the values gds (option A) and icwb (option B), is used to switch between different layout file reading options. The stretches are applied to all layers with the command:

```
icwb stretch name= "MOS_L" value=@<(lgate-0.030)>@
```

Sentaurus Process

The process flow in this project represents a 32 nm technology node. The major steps to create NMOS and PMOS devices are:

1. Well definition.
2. High-k gate oxide deposition.
3. Metal-gate definition.
4. Halo implantation.
5. Source–drain extension formation (LDD).
6. Highly doped source–drain formation (HDD).
7. Spike anneal and laser annealing.
8. Metallization.

The basic model selection for the complete process simulation consists of:

- Three-stream or pair diffusion model.
- Three-phase segregation model.
- Mechanical stresses are computed using different material models for different materials, that is, the anisotropic elasticity model for silicon and the plastic material model for polysilicon.
- Stress induced by lattice mismatch for Si:C pockets and SiGe pockets.
- Stress effects on dopant diffusion and activation.
- Silicidation model.

The process flow is simulated with the Advanced Calibration set of models and parameters. The use of Advanced Calibration ensures accurate and optimized results for many processes in device fabrication such as ion implantation, laser annealing, oxidation, ultrashallow junction formation, surface dose loss, and halo dopant diffusion.

The Advanced Calibration model settings are defined at the beginning of the process simulation using:

```
AdvancedCalibration
```

Splits in Sentaurus Workbench are used to separate different parts of the simulations: After the processing of certain steps, the simulation stops and the intermediate structure is saved. The subsequent processing steps are performed in a new simulation and the previous simulation results are reloaded. The TDR mode is used for saving and reloading, which ensures that all Sentaurus Process settings are automatically transferred from the previous simulation.

The unified coordinate system (UCS) is activated by using the following command in the Sentaurus Process command file:

```
math coord.ucs
```

In the Sentaurus Process coordinate system, the x-axis points down into the device.

The bounding box of the selected simulation domain is stored in the Tcl variables Ymin and Ymax:

```
set Ymin [icwb bbox left]
set Ymax [icwb bbox right]
```

The extent of the simulation domain in the vertical direction is given explicitly:

```
set Xbot 10
set Xtop 0
```

These variables can be used to create the initial structure:

```
line x location= $Xtop tag= top
line x location= $Xbot tag= bottom
line y location= $Ymin tag= left
line y location= ($Ymax-$Ymin)/2.0 tag= right

region Silicon substrate xlo= top xhi= bottom \
  ylo= left yhi= right
```

Sentaurus Mesh creates the meshes. It is the default meshing engine of Sentaurus Process. With Sentaurus Mesh and MGOALS (an internal module of Sentaurus Process), bulk refinement and interface refinements are controlled. The following command allows you to define the remeshing strategy using MGOALS and to set parameters for the MGOALS level-set operation:

```
mgoals on min.normal.size= 0.001/$f \
  normal.growth.ratio= 2.0
refinebox interface.materials= Silicon
```

where the parameter `f` scales the mesh resolution as discussed in Sentaurus Process on page 3.

A set of refinement boxes is used to define the bulk mesh. The location of the refinement boxes is parameterized with geometric and layout parameters to make the setup robust. The parameterized refinement boxes and their locations are selected using the ICWBEV Plus command interface. One such refinement box to capture the halo implantation is:

```
refinebox name= HALO mask= Active \
  extend= 0.0 extrusion.min= 0.0 \
  extrusion.max= $HaloDepth \
  xrefine= "0.002/$f 0.005/$f" \
  yrefine= "$PolyPitch/(16.0*$f)" Silicon add
```

The project makes use of the symmetry of the structure and simulates only half of the device structure. The halo implantation step, however, is performed at a shallow angle, such that for a small gate length, ions that, for example, enter on the source side may end up on the drain side. You can use the following global switch to provide unified treatment in 2D analytic implantations:

```
pdbSet ImplantData Extrude 1
```

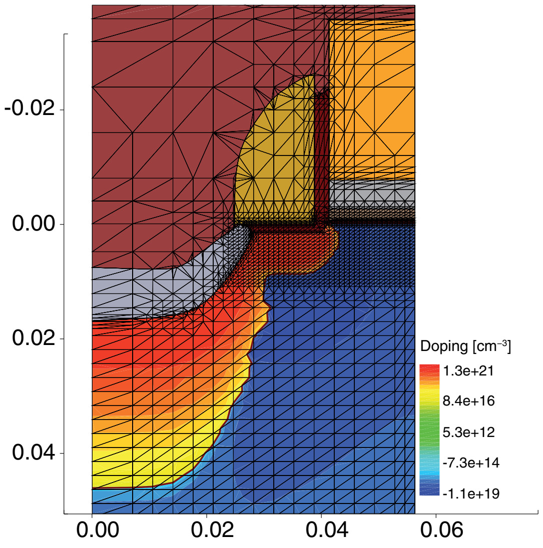


Figure 2 Magnification of gate–drain area of 30 nm length NMOS device as simulated by Sentaurus Process; the net doping concentration in various regions is shown

In this case, the 2D structure is extruded in pseudo-3D mode and performs a 3D-like implantation.

A metal gate is used here to avoid the poly depletion effect. The metal-gate processing steps are compatible with those of the high-k dielectric. Selecting the most appropriate material for both NMOS and PMOS poses a considerable challenge in itself. The high-k/metal gate in Sentaurus Process is initialized with the commands:

```
deposit HfO2 thickness= $THF<um>
mater add name= AlN new.like= Nitride
mater add name= TiN new.like= Nitride

#if [string match @CUT@ "SIM2N"]
deposit TiN thickness= 0.005
#else
deposit AlN thickness= 0.005
#endif
```

Figure 2 shows the net doping profile and mesh points after process simulation.

Stress Simulation Setup

The 32-nm CMOS technology continues to use mobility enhancement techniques such as tensile and compressive stress liner and embedded stressor (eSiGe) for PMOS, and Si:C source–drain (S/D) and stress memorization techniques for NMOS. It is very important to optimize the device structure to add a sufficient amount of stress to the channel region.

The stress-related `pdb` parameters are switched on using the following commands, which are defined at the very beginning of the process commands in the `sprocess.fps` file:

```
pdbSetBoolean Mechanics StressHistory 1
pdbSet Silicon Mechanics Anisotropic 1
```

Sentaurus Process can simulate thermal-induced and process-induced stress during structural modification. Therefore, the definition of intrinsic stress after the topography operation is very important. In the 32-nm CMOS flow, the process setup requires the deposition or growth of material with intrinsic stress.

Dual stress liner (DSL)–induced channel stress is implemented differently. The DSL material is deposited with the negative pressure field to produce isotropic tensile stress:

```
doping name= Stress_TensDSL field= Pressure \
  depths= {-0.3 0.01} values= \
  { -1*$STRESS_TENSILE_NMOS_DSL_Pa \
  -1*$STRESS_TENSILE_NMOS_DSL_Pa }

deposit material= {Oxynitride} \
  doping= {Stress_TensDSL} type= isotropic rate= 1.0 \
  time= 0.075 steps= 10
```

The Si:C stressor in the S/D extension regions of the NMOS transistors is formed by using the epitaxial growth process, but the mole fraction–dependent stress model must be defined properly:

```
temp_ramp name= epi_SiC temperature= 550<C> \
  t.final= $epi_temp<C> time= 1.0<min> \
  thick= 0.76*$epi_thick<um> \
  epi.doping= {Boron= 1.5e17 Carbon= 2.5e21} \
  Epi epi.doping.final= \
  {Boron= 8.0e16 Carbon= 2.5e21} epi.layers= 10 \
  epi.model=1

diffuse temp_ramp= epi_SiC info= 1
```

The lattice mismatch–induced mechanical stress model for SiGe is activated by default in Sentaurus Process. The condition is slightly different for Si:C.

To describe mole fraction–dependent stress in Si:C, you must describe the strain profile as a piecewise linear function. Carbon is smaller than silicon, and it has an opposite effect on silicon compared to germanium. The `Conc.Strain` parameter must be changed before the thermal steps as:

```
pdbSetDoubleArray Silicon Carbon Conc.Strain \
  {0 0 1 -0.432}
```

The final mechanics computation is performed by invoking the `UpdateStrain` command followed by a small thermal step with short duration to update the mechanics:

```
pdbSetDouble Silicon Mechanics TopRelaxedNodeCoord \
  0.15e-4
pdbSetBoolean Silicon Mechanics UpdateStrain 1
diffuse stress.relax time= 1e-10 temp= 600
```

To generate accurate stress in the channel of a PMOS, the two-step recessed SiGe process is implemented to form compact SiGe pockets. Successive etching and isotropic deposition steps are used to create the dome and, finally, the two-step strained deposition process is used to obtain the required amount of germanium and stress in the pocket region. Initially, you define the doping field along the required depth. Then, isotropic deposition of silicon material with the correct doping field is performed. The associated strained-layer deposition for the SiGe pocket is:

```
doping name= Epi1B field= Boron \
  values= {1e19 2.0e20} depths= {0.0 0.3*$epi_thick}
doping name= Epi1As field= Arsenic \
  values= {2e18 1e18} depths= {0.0 0.3*$epi_thick}
doping name= Epi1Ge field= Germanium \
  values= {1.5e22 1.5e22} \
  depths= {0.0 0.3*$epi_thick}

deposit material= {Silicon} type= isotropic \
  time= 1.0<min> rate= 0.4*$epi_thick \
  selective.materials= {Silicon} \
  doping= {Epi1B Epi1As Epi1Ge} \
  steps= 1 Strained.Lattice
```

Similarly, another step with the required doping field is needed to form the complete SiGe pocket. Before the strained deposition process, you can update some default process pdb parameters to activate the model:

```
pdbSet Mechanics StrainedDeposition 1
```

Finally, the mechanics and substrate stress is updated for diffusion with a very small time step.

Sentaurus Process can simulate stress memorization using the elastoplastic and viscoelastic models.

To switch on the plastic material model, you must set the yield stress, the thermal coefficient, and the isotropic hardening modulus for polysilicon using the commands:

```
pdbSetDouble PolySilicon Mechanics ThExpCoeff 20e-6
pdbSetDouble PolySilicon Mechanics FirstYield 5.0e10
pdbSetDouble PolySilicon Mechanics \
  Hardening.Modulus.Isotropic 1.0e10
```

The following flag must be switched on to activate the plastic deformation simulation before S/D thermal activation:

```
pdbSetBoolean PolySilicon Mechanics IsPlastic 1
```

The plasticity model must be switched off when the thermal steps are finished. The final stress distribution (S_{yy}) in the 2D device structures for the NMOS and PMOS devices are shown in Figure 3.

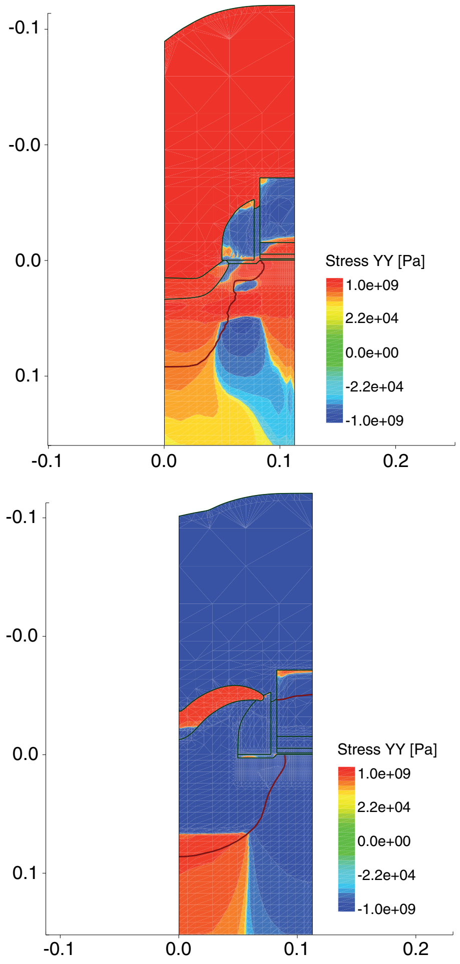


Figure 3 Longitudinal component of stress (S_{yy}) distribution for (*top*) NMOS device and (*bottom*) PMOS device

Laser Annealing Simulation

Laser annealing is an alternative to the spike-RTA process to achieve superactivation of the dopants with no extra diffusion. To simulate the phenomena of laser-induced high activation and less diffusion of the dopant under the channel, the advanced features of Sentaurus Process have been used in this project. The process temperature and boundary conditions during the ramp-up and ramp-down steps are important parameters to solve the heat equation in laser annealing simulations. To define the laser source and the temperature profile, and to update the heat generation rate, the parameters must be set correctly:

```
pdbSetDouble Heat Fluence 8.6
pdbSetDouble Heat Pulse 0.11
pdbSetBoolean Heat UpdateHeatRate 1
```

The heat boundary condition, the light source definition for the laser pulse, and the thermal properties of the material are described in the `Laser.fps` file, which is called before the laser simulation:

```
source Laser.fps
```

Finally, the laser anneal simulation is activated using the `diffuse` command:

```
diffuse laser temp= $LASER_AMBIENT_TEMP_degC \
  time= $LASER_DURATION_ms<ms> stress.relax \
  info= 2 write.temp.file= n@node@_LATemp
```

Implementing Silicide Contact

Silicides in the S/D of MOSFET devices attract attention as contacts and interconnect materials in silicon microelectronic technology. The growth process of nickel-silicide (NiSi) material is simulated in this project. NiSi is assumed to form when silicon atoms react with nickel atoms. The simulation logic is that an initial layer of NiSi (1.5 nm) is specified using the command:

```
pdbSet Grid NativeLayerThickness 1.5e-7
```

The silicidation process can be influenced by the presence of oxygen near any metal–silicon interface. This oxygen is assumed to enter the silicide at interfaces with SiO_2 and to diffuse in the silicide according to Fick’s law and to change the silicidation growth mechanism. The process can be suppressed using:

```
pdbSet NiSi Silicon ORS 0
```

The final reaction rate can be modified or optimized using:

```
pdbSetDouble NickelSilicide_Silicon Silicon \
  Expansion.Ratio 1.0
pdbSetDouble NickelSilicide_Silicon Silicon \
  Density.Grow 2.5e22
```

During the silicidation process, two or more materials such as oxide, silicon, and silicide may come together and try to move inadvertently due to material consumption around a material node. This is referred to as the *triple-point movement*, which can produce structural deformation. To control the movement, a retardation factor around the triple point is applied in this project and is controlled by the parameters:

```
pdbSetDouble NickelSilicide_Silicon /
  SilicidationTripleDistance 20e-7
pdbSetDouble NickelSilicide_Silicon /
  SilicidationTripleFactor 0
```

To switch the triple-point control on and off, use the flag:

```
pdbSet Mechanics SilicidationCorrection {1 | 0}
```

All these important parameters are sourced from the `Nisilicide.fps` file, while the final silicidation is performed using:

```
source Nisilicide.fps
deposit Nickel type= isotropic thickness= 0.015
diffuse time= 0.5<s> temp= 500<C> minT= 350<C> \
  stress.relax
strip Nickel
```

Preparing Structures for Device Simulations

At the end of the process simulation flow, Sentaurus Process prepares the structures for the subsequent device simulations with Sentaurus Device. To simplify the structure and to introduce the metal-gate contact, the polysilicon gate is removed in the `Mesh` split. Furthermore, the top and bottom parts of the structures are truncated for fast device simulation by reducing the node points:

```
transform cut min= {-0.5 -10 -10} max= {0.5 1.0 1.0}
```

A deep substrate is required during process simulations to account for the fast and deep diffusion of point defects, but it is unnecessary for device simulations. In addition, meshing requirements for process and device simulations are different. For example, for process simulations, it is convenient either to use a fixed meshing strategy or to adapt the mesh to the gradients of individual dopant and cluster profiles. For device simulations, these profiles are not of particular interest. It is more useful to refine on the net active doping concentration. For this remeshing, first, all the previously defined refinement boxes are removed:

```
refinebox clear
refinebox !keep.lines
line clear
```

Second, the adaptive meshing defaults are altered to suppress refining profiles irrelevant for device simulations:

```
pdbSet Grid AdaptiveField Refine.Abs.Error 1e37
pdbSet Grid AdaptiveField Refine.Rel.Error 1e10
pdbSet Grid AdaptiveField Refine.Target.Length 100.0
pdbSet Grid Adaptive 1
```

The meshing engine is configured to create high-quality mesh elements best-suited for device simulations:

```
pdbSet Grid SnMesh DelaunayType boxmethod
```

The adaptive meshing based on the gradient of the net active doping concentration is activated with the `refine.fields` option of the `refinebox` command, for example:

```
refinebox name= DA mask= Active \
  extend= 0.01 extrusion.min= -0.05 \
  extrusion.max= 0.2 refine.fields= {NetActive} \
  def.max.asinhdiff= 0.5 \
  refine.max.edge="0.01/$fd $PolyPitch/(8*$fd)" \
  refine.min.edge= "0.002/$fd $PolyPitch/(32*$fd)" \
  adaptive $Sub add
```

The channel–gate oxide interface is resolved with a fine mesh for device simulation to resolve the inversion layer:

```
refinebox name= DeviceIF mask= Device_Channel \
  extend= 0.01 extrusion.min= -0.001 \
  extrusion.max= 0.002 min.normal.size= 0.0002/$fd \
  normal.growth.ratio= 1.1 \
  interface.materials= {Silicon Oxide} $Sub add
```

Before assigning the contacts, the structures are reflected:

```
transform reflect right
```

The layout-driven contact placement command `icwb.contact.mask` is used to create the contact for the subsequent device simulation. The command serves as an interface between the ICWBEV Plus TCAD layout and the Sentaurus Process `contact` command. The corresponding commands for NMOS contacts are:

```
icwb.contact.mask layer.name= "Drain_N1" \
  name= drain point NickelSilicide \
  x= $NMOS_recess+0.0025 adjacent.material= $Sub

icwb.contact.mask layer.name= "Source_N1" \
  name= source point NickelSilicide \
  x= $NMOS_recess+0.0025 adjacent.material= $Sub

icwb.contact.mask layer.name= "Gate_N1" name= gate \
  point TiNitride x= -$THF-$Tox-0.004 replace
```

Device Simulation Using Sentaurus Device, and Extraction and Visualization Using Sentaurus Visual

Sentaurus Device is used to simulate the drain current as a function of the gate voltage at a low-drain bias and a high-drain bias (I_d–V_{gs}) as well as to simulate a family of drain current curves as a function of the drain voltage (I_d–V_{ds}) for different values of the gate bias. Furthermore, a small-signal AC analysis is performed to show the total-gate, gate–contact, and gate–body capacitance as a function of the gate voltage (C–V). In the Sentaurus Workbench tool flow, each instance of Sentaurus Device is followed by an instance of the visualization tool Sentaurus Visual, which plots the corresponding I–V or C–V characteristics, and extracts relevant electrical parameters.

Sentaurus Device performs the device simulations using the Advanced Calibration set of models and parameters for 32 nm Si/SiGe CMOS with high-k metal gate technology. The influence of mechanical stress on NMOS and PMOS device performances is also included in these models. To use the Advanced Calibration of Sentaurus Device, you can look up the parameter files and the command files for specific device types and technology requirements as described in the *Advanced Calibration for Device Simulation User Guide* [1].

To start with Advanced Calibration for CMOS, the parameter files `SiliconGermaniumc100.par` and `Siliconc100.par` are used in this project from the following location:

```
$STROOT/tcad/$STRELEASE/lib/sdevice/MaterialDB/
```

The Advanced Calibration set of models for the simulations of 32 nm CMOS device characteristics is defined in the `Physics` section of the command file. The recommended models are:

- Drift-diffusion transport model
- Low-field mobility degradation at high-k–oxide interfaces
- Strained band structure and electrostatic-related models
- Density-gradient quantum corrections
- Strained low-field and high-field mobility models
- Surface orientation dependency of the inversion and accumulation layer mobility model

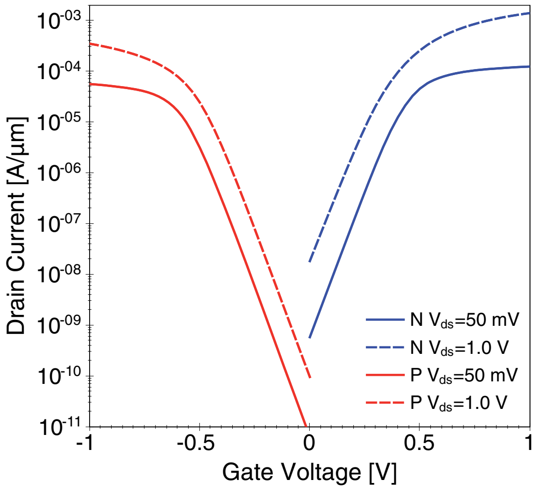


Figure 4 Drain current as a function of gate voltage for 30 nm gate length PMOS (red) and NMOS (blue) devices simulated with Sentaurus Device; lower curves (solid lines) are for a drain bias of 50 mV and upper curves (dashes) are for 1.0 V

Figure 4 shows the I_d–V_g characteristics for the 30 nm device simulated at V_{ds} = 50 mV, 1.0 V, with stress.

Similarly, the output characteristics at two gate biases (`Vg1` and `Vg2`) for the NMOS and PMOS are shown in Figure 5 on page 10.

To obtain the C–V characteristics, the AC analysis is performed at 1 MHz during a gate voltage sweep from $-1.5 \cdot \texttt{Vdd}$ to $+1.5 \cdot \texttt{Vdd}$. Figure 6 shows the C–V curves for the 30 nm NMOS device.

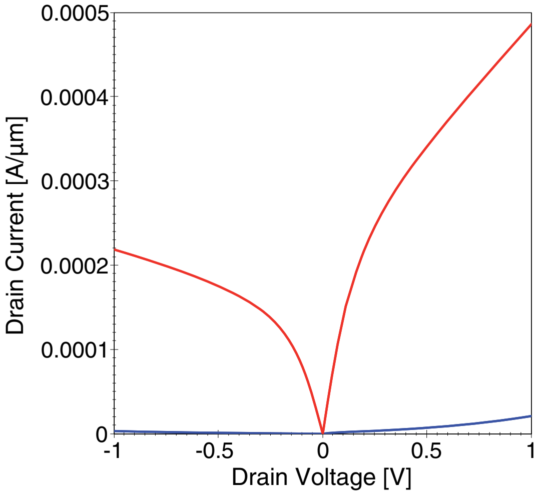


Figure 5 Drain current as a function of drain voltage for 30 nm gate length PMOS (*left*) and NMOS (*right*) devices simulated with Sentaurus Device; gate bias for curves are Vg1 (blue) and Vg2 (red)

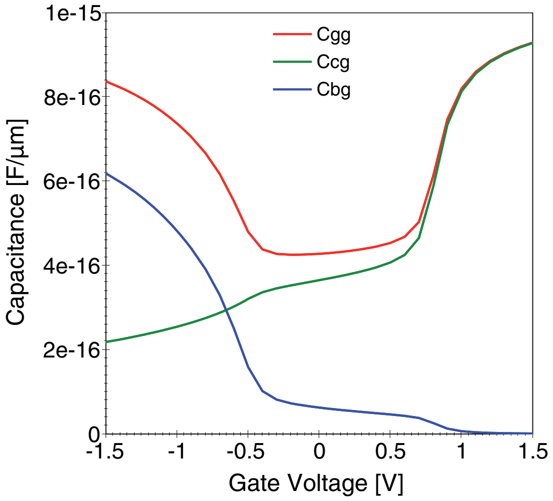


Figure 6 Total gate (Cgg, red), gate-contacts (Ccg, green), and gate-body (Cbg, blue) capacitance as a function of gate voltage for 30 nm gate length NMOS device simulated with Sentaurus Device

The device simulation and extraction setup used here is identical to the one discussed in *Sentaurus Technology Template: CMOS Characterization*. Refer to the documentation for that project for details about the device simulations, visualization, and parameter extraction.

References

[1] *Advanced Calibration for Device Simulation User Guide*, Version J-2014.09, Mountain View, California: Synopsys, Inc., 2014.