

目录

Sentaurus TCAD 培训.....	1
工具简介.....	2
A. TCAD 是什么? (10 分钟)	2
A.1 TCAD 与半导体工业.....	2
A.2 工艺模拟.....	2
A.3 器件模拟.....	2
B. TCAD 包含哪些工具? (20 分钟)	3
B.1 Sentaurus Workbench.....	3
B.2 Ligament.....	4
B.3 Sentaurus Process.....	4
B.4 Sentaurus Structure Editor (SDE)	5
B.5 Mesh and Noffset3D.....	5
B.6 Sentaurus Device.....	6
B.7 Tecplot SV.....	6
B.8 Inspect.....	7
B.9 Calibration Kit.....	7
SWB.....	8
A. Getting Started (15 分钟)	8
A.1 概述.....	8
A.2 启动 SWB.....	8
B. 运行工程 (30 分钟)	9
B.1 加载 SWB 工程.....	9
B.2 改变树的显示属性.....	10
B.3 清空工程目录.....	10
B.4 运行工程.....	10
B.5 查看输出结果.....	11
C. 创建工程 (30 分钟)	11
C.1 设置工具流.....	11
D. 组装多个实验 (30 分钟)	12
D.1 添加参数.....	12
D.2 设置多个实验.....	13
D.4 创建方案.....	14
D.5 裁剪工程树.....	15
D.6 添加变量.....	15
Ligament.....	16
A. Getting Started 10 min.....	16
A.1 简介.....	16
A.2 Ligament 流程编辑器.....	16
A.3 Ligament 版图编辑器.....	16
A.4 Ligament Translator.....	16
B. Ligament 流程编辑器 30 min.....	16

B.1 启动 Ligament 流程编辑器.....	16
B.2 开始装配一个新的工艺流程.....	17
B.4 改变宏调用的参数.....	17
B.7 用户定义的变量.....	17
B.8 转换和语法检查.....	18
C. Ligament 版图编辑器 30 min.....	19
C.1 启动 Ligament 版图编辑器.....	19
C.4 保存版图.....	19
C.6 定义模拟区域.....	19
C.7 使用版图.....	20
D. 在 SWB 中运行 Ligament 30 min.....	20
D.1 在 SWB 中启动 Ligament 工具.....	20
D.2 导入 Ligament 文件.....	20
D.5 Ligament 版图编辑器中的版图参数化.....	21
D.6 SWB 中的参数化.....	21
D.7 Ligament 工作区.....	21
D.8 SWB 的预处理和 Ligament 转换.....	22
Sentaurus Process.....	23
A. Getting Started 15min.....	23
A.1 简介.....	23
A.2 Sprocess 使用的文件类型.....	23
A.3 启动 Sprocess.....	23
B. 一维工艺模拟 45min.....	24
B.1 简介.....	24
B.2 定义初始一维网格.....	24
B.3 定义初始模拟域(domain).....	24
B.4 初始化.....	24
B.5 设置 MGOALS 网格化策略.....	24
B.6 Growing Screening Oxide.....	24
B.7 测量氧化物的厚度.....	25
B.8 Depositing Screening Oxide.....	25
B.9 Tcl 控制语句.....	25
B.10 注入.....	25
B.11 Saving the as-implanted Profile.....	26
B.12 热退火, Drive-in, Activation, and Screening Oxide Strip.....	26
C. 二维工艺模拟 45min.....	27
C.1 简介.....	27
C.2 定义初始的二维网格.....	27
C.3 模拟域与初始化.....	27
C.4 B 的注入.....	28
C.5 生长栅氧.....	28
C.6 制作多晶硅栅.....	28
C.7 多晶硅重氧化.....	28
C.8 保存 Snapshots.....	29
C.9 在 LDD 和 Halo 注入之前进行网格重定义.....	29

C.10 LDD 和 Halo 注入.....	30
C.11 形成边墙.....	30
C.12 在源/漏注入之前进行网格重定义.....	30
C.13 源/漏注入.....	31
C.14 Contact Pads.....	31
C.15 保存整个结构.....	31
C.16 提取一维分布.....	32
D. 定义模型和指定参数 30min.....	32
D.1 属性数据库浏览器.....	32
D.2 在输入文件中改变参数.....	33
E. 采用全定制的校准文件 30min.....	33
E.1 简介.....	33
E.2 全局设置.....	33
F. 采用高级校准 30min.....	33
F.1 简介.....	33
F.2 激活高级校准.....	33
G. 工艺模拟与 SWB 和 Ligament 30min.....	34
G.1 简介.....	34
G.2 参数化版图.....	34
G.3 定义网格.....	35
H. Custom Models with Alagator 45min.....	35
H.1 简介.....	35
I. Special Focus: 槽刻蚀 35min.....	36
I.1 简介.....	36
I.2 Initialization.....	36
I.3 生长 Pad Oxide.....	36
I.4 淀积氮化层.....	37
I.5 STI 光刻.....	37
I.6 浅槽刻蚀.....	37
I.7 生长衬垫氧化物.....	38
I.8 TEOS 的淀积与 CMP.....	38
I.9 氮化物的剥离/反射.....	39
J. Special Focus: Meshing with MGOALS(15 分钟).....	39
J.1 简介.....	39
J.2 Initialization.....	39
J.3 MGOALS Remesh.....	40
J.4 MGOALS 网格优化框.....	40
K. Special Focus: 3D 工艺模拟 (Sprocess 与 sde 结合) (30 min).....	41
K.1 简介.....	41
K.2 各项异性刻蚀.....	41
K.3 各向同性淀积.....	42
K.4 Multimaterial Etching.....	43
K.5 Reflect and Clip.....	44
SDE (结构编辑器)	46
A. Getting Started (15 分钟)	46

A.1 简介.....	46
A.2 启动 SDE.....	46
A.4 输入/输出文件类型.....	46
A.5 创建一个简单的结构.....	47
B. 生成二维边界 (45 分钟)	48
B.1 简介.....	48
B.2 重新初始化 SDE.....	48
B.3 Exact Coordinates 模式.....	48
B.4 选择材料.....	48
B.5 选择默认的布尔表达式.....	48
B.6 创建矩形区域.....	49
B.7 创建 Single-Lumped 区域.....	49
B.8 圆化边缘.....	49
B.9 定义接触.....	50
B.10 在已存在的边缘处设置接触.....	50
B.11 添加顶点.....	51
B.12 定义一个区域为接触.....	52
B.13 对区域进行重命名.....	52
B.14 保存模型.....	52
C. 生成掺杂分布 (20 分钟)	53
C.1 定义材料的掺杂浓度为常数.....	53
C.2 定义区域的掺杂浓度为常数.....	53
C.3 定义解析的掺杂分布.....	53
C.4 保存模型(见图 4).....	54
D. 生成网格 (20 分钟)	55
D.1 简介.....	55
D.2 定义区域的网格化策略.....	55
D.3 定义优化窗口.....	55
D.4 定义优化窗口的网格化策略.....	56
D.5 在优化窗口中定义一个 Multibox 网格化策略.....	56
D.6 保存模型.....	56
D.7 为器件结构产生网格.....	56
E. 脚本和参数 (30 分钟)	57
E.1 简介.....	57
E.2 Scheme 基础.....	57
E.3 定义简单的变量和数据类型.....	57
E.5 数学操作符.....	58
E.6 数学表达式.....	58
Noffset3D.....	59
A. Getting Started 10 min.....	59
A.1 Overview.....	59
A.2 Starting Noffset3D.....	59
B. Using Noffset3D in 2D 40 min.....	59
B.1 Strategies of Meshing.....	59
B.2 Main Parameters for Meshing.....	59

C. Using Noffset3D in 3D 40 min.....	60
C.1 Main Algorithm.....	60
Sentaurus Device.....	61
A. 基础 (15 分钟)	61
A.1 简介.....	61
A.2 输入命令文件.....	61
A.3 参数文件.....	65
B. 载流子传输模型 (45 分钟)	65
B.1 简介.....	66
B.2 漂移-扩散传输.....	66
B.3 热力学传输.....	67
B.4 流体力学传输.....	70
B.5 密度梯度传输.....	72
B.6 Monte Carlo 传输.....	74
C. 混合模式模拟 (45 分钟)	77
C.1 简介.....	77
C.2 混合模式模拟.....	77
C.3 瞬态扫描.....	78
C.4 小信号 AC 分析.....	80
D. 一个较复杂的例子 (30 分钟)	81
D.1 MOSFET 的击穿模拟.....	81
D.2 电流边界条件.....	83
G. 在 SWB 中运行 SDevice (25 分钟)	85
G.1 简介.....	85
G.2 File Section.....	85
G.3 使用 SWB 参数.....	86
G.4 算术表达式.....	87
G.6 参数文件的参数化.....	87

Sentaurus TCAD 培训

- 1、工具简介：30 分钟
- 2、Sentaurus workbench：一个可视化的集成环境（2 小时）
- 3、Ligament：TCAD 工艺模拟的一个通用接口。可以在更高的抽象层次上设置和执行 TCAD 工艺模拟。
（译者注：如果没有 Ligament，需要手工编写用于工艺模拟的命令文件；有了 ligament 后，可以采用 ligament 以比较简单直观的方式设置工艺模拟的步骤。不过，在 ISETCAD10 中，ligament 的功能还不够强大，使用 ligament 只能得到一个工艺模拟命令文件的模板，需要经过手工修改后方可用于真正的工艺模拟。）（1 小时 40 分钟）
- 4、Sentaurus Process：一个完整的高度灵活的多维工艺模拟器。（5 小时 20 分钟）
- 5、Sentaurus Structure Editor：一个二维和三维器件编辑器以及三维工艺仿真器。基于 CAD 技术，具有一个功能强大的 GUI，并且可以全部采用脚本实现。（4 小时 15 分钟）
- 6、Noffset3D：一个网格生成器，在二维时用于生成三角形和矩形，在三维时生成四面体。面向 Sentaurus Device 和 Sentaurus Process 等采用方框法（box method）离散技术的模拟器。（1 小时 30 分钟）
- 7、Sentaurus Device：器件模拟器，用于模拟半导体器件的电、热和光特性。是业界领先的器件模拟器，能够处理一维、二维和三维的几何结构，以及混合模拟（同时模拟集约模型和数值器件）。（注：ISETCAD10 中的混合模拟的集约模型只支持 level3，会损失一些精度。）（4 小时）
- 8、Tecplot SV：用于查看模拟和实验数据的二维和三维绘图软件。（2 小时）
- 9、Inspect：一个 x-y 数据的绘图和分析工具，例如半导体器件的掺杂分布和电特性。（1 小时 45 分钟）
- 10、TCL：大部分 Synopsys 的工具均采用 TCL（Tool Command Language），掌握一些关于 TCL 的基本知识是有益的。（1 小时 20 分钟）

注：拟举办一个为期一周（5 天）的培训班。面向 XX 组新进入的人员和 XX 老师研究器件的学生。

工具简介

A. TCAD 是什么？（10 分钟）

目标：对 TCAD 进行简要介绍，明确这样一个问题：谁使用 TCAD？

A.1 TCAD 与半导体工业

TCAD 亦即 Technology CAD，指的是采用计算机模拟开发和优化半导体工艺技术和半导体器件。TCAD 模拟工具解算基本的物理偏微分方程，例如扩散和传输方程。这种底层的物理方法使得 TCAD 模拟可以产生具有预测价值的结果（译者注：主要指精度可以与试验结果相近）。

因此，在开发新的半导体器件或工艺时，有可能采用 TCAD 计算机模拟代替（注：采用“部分代替”这个词或许更合适）昂贵而且耗时的测试晶圆（test wafer）。

TCAD 模拟在半导体工业界被广泛使用。随着技术变得越来越复杂，半导体工业界愈发依赖于 TCAD 以降低成本，并加速研发进程。另外，半导体制造商采用 TCAD 进行成品率分析，亦即：监测、分析和优化其 IC 工艺流程，分析工艺起伏的影响。

TCAD 包含两个主要的分支：工艺模拟和器件模拟。

A.2 工艺模拟

在工艺模拟时，基于各种物理方程（不同的工艺步骤对应不同的物理方程）模拟诸如刻蚀、淀积、离子注入、热退火和氧化等各种工艺步骤。被模拟的晶圆被离散化，表现为一种有限元结构（见图 1）。例如，在模拟热退火时，在该网络上解算每种杂质的扩散方程。对于氧化模拟，基于氧气的扩散、边角处的机械应力等模拟 SiO_2 的生长。

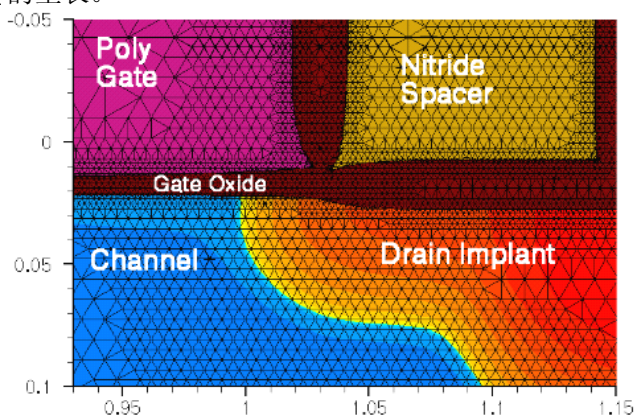


图 1 NMOSFET 栅漏交汇处有限元网格的放大图

A.3 器件模拟

器件模拟可以被想象为半导体器件（如晶体管或二极管）电特性的虚拟测量。器件被描绘成离散化的有

限元结构。器件的每个网格点都有相应的性质与之关联，例如材料的种类和掺杂浓度。器件模拟其实就是计算每一个网格点的载流子浓度、电流密度、电场、产生和复合速率、等等（见图 2）。

电极被描绘成多个区域（are represented as areas），在每个区域施加边界条件，例如电压。器件模拟器解算泊松方程和载流子连续性方程（也可能是其它方程）。在解算完这些方程之后，可以提取触点（contacts）处的电流（见图 3）。

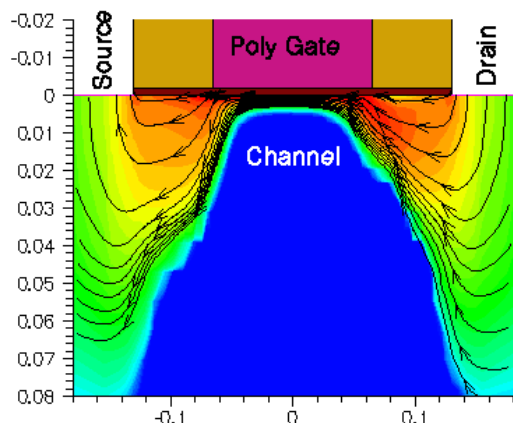


图 2 0.13μm NMOSFET 在 $V_{gs}=1.5V$, $V_{ds}=3.0V$ 条件下的电流密度线，不同颜色代表电流密度的大小。

Q: 谁使用 TCAD?

A: 工艺制造商、器件设计者、其它需要研究工艺和器件的人员（如：抗辐照加固领域的科技工作者）

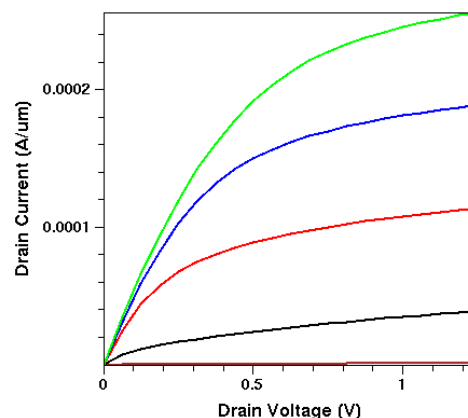


图 3 50nm NMOSFET 的输出特性
($V_{gs}=0.25, 0.5, 0.75, 1.0, 1.25V$)

B. TCAD 包含哪些工具？（20 分钟）

目标：对 TCAD 的各种工具进行简要介绍。

B.1 Sentaurus Workbench

一个可视化的集成环境，其直观的 GUI 可用于设计、组织和运行模拟。各种模拟以工程的形式进行组织。一个完整的模拟流程通常包括多个工具，例如工艺模拟器 Sentaurus Process，网格化工具 mesh，器件模拟器 Sentaurus Device，绘图和分析工具 inspect。

Sentaurus Workbench 自动管理从一个工具到另一个工具的信息流。包括用户输入文件的前处理、参数化工程、工具的设置和执行以及结果的查看。

Sentaurus Workbench 允许用户定义参数和变量以运行复杂的参变量分析。采用数学和逻辑表达式对模拟的输入文件进行动态前处理。模拟的结果可输入到统计分析软件和电子表格软件。

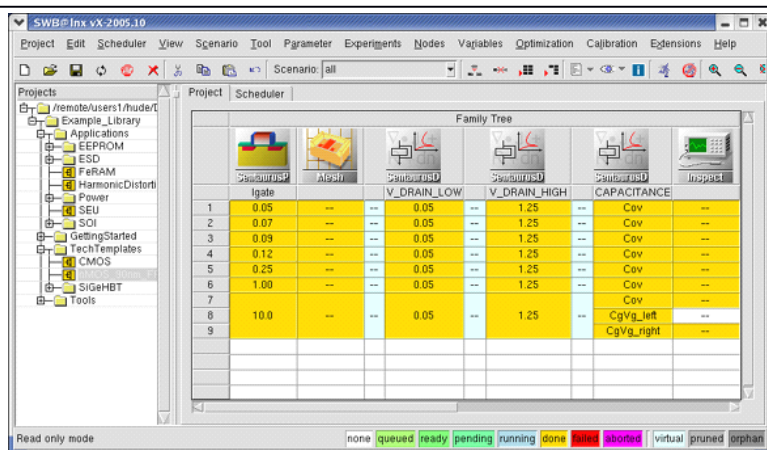


图 1 加载工程后的 SWB GUI。左边为工程浏览器，中间为工程的工具流，工具流的下面是各种参数定义。

B.2 Ligament

Ligament 是 TCAD 工艺模拟的一个通用接口。利用 Ligament，可以在一个更高的抽象层次上设置和执行 TCAD 模拟。自动产生模拟程序的输入文件。Ligament 被集成在 SWB 中，可以与所有的 Sentaurus 模拟程序协同工作。开放式的架构允许用户对命令和模拟转换模块进行定制化的修改和添加。

Ligament 流程编辑器

Ligament 流程编辑器提供一个方便的 GUI 用于工艺流程的创建和编辑。采用宏（包括用户定义的局部宏和 Ligament 工具库中定义的全局宏）装配一个工艺流程。Ligament 采用 SPR（Simple Process Representation）语言，独立于任何特定工艺模拟器的语法。

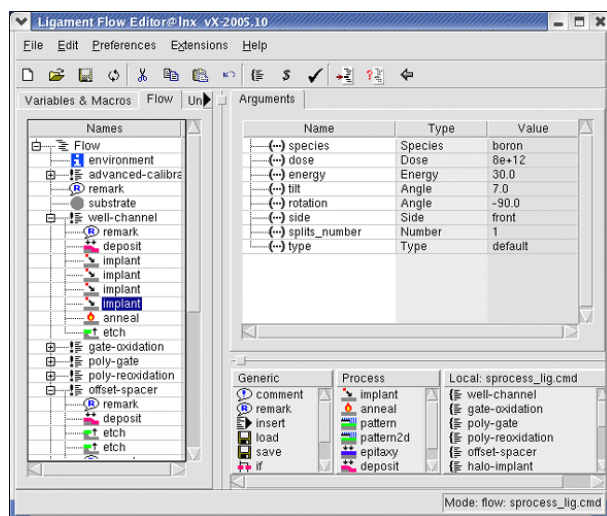


图 2 Ligament 流程编辑器

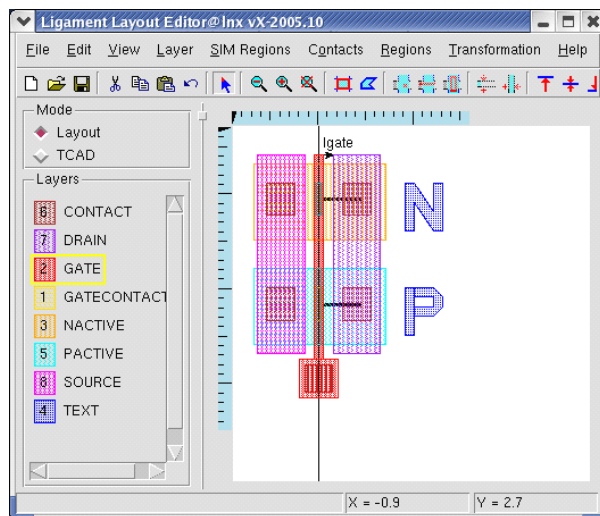


图 3 采用 Ligament 版图编辑器绘制的参数化试验版图。两条水平粗线定义了 N 管和 P 管二维区域的割线。垂直细线用于定义参数，版图可以根据参数 Igate 进行水平方向的拉伸（译者注：类似于 pcell）。

Ligament 版图编辑器

Ligament 版图编辑器提供一个 GUI 用于创建和编辑版图。Ligament 版图编辑器是 EDA 版图工具和 TCAD 之间的一个接口，可用于定义一维、二维或三维模拟区域或者进行版图的参数化拉伸（例如在 Vt roll-off 模拟时）。Ligament 版图编辑器支持 GDSII 和 GIF 格式的外部版图。

B.3 Sentaurus Process

Sentaurus Process 是一个完整的和高度灵活的多维工艺模拟环境。采用验证过的 (proven) 校准方法面向大量的最新实验数据进行校准, Sentaurus Process 可以为现代 Si 工艺和非 Si 工艺提供空前的预测能力。Sentaurus Process 基于最初由 Florida 大学开发, 现在广泛使用的软件。Synopsys 强化了该软件的一些特性, 将 Sentaurus Process 从一个广泛使用的研究工具转换成一个快速且健壮的 (robust) 工业软件。与 Synopsys 的工艺模拟器 Dios 完全兼容。

增强的部分包括: 一个与 SDE 的接口, 用于三维刻蚀和淀积; 一个模拟 SiGe 和应变 Si 的晶格失配模型; 基于 Crystal-TRIM 的 Monte Carlo 注入; 先进的校准注入表; 解析的注入和损伤模型; 代表最新发展水平的扩散模型。另外, 还有一个基于 levelset 方法的被称为 MGOALS 的网格化模块。MGOALS、高质量的各向异性网格和高性能的求解程序 PARDISO 结合在一起, 形成了优秀的效率和鲁棒性。

Sentaurus Process 中使用的 Alagator 脚本语言允许用户快速开发新的扩散模型, 使得 Sentaurus Process 成为一个十分方便的校准工具。

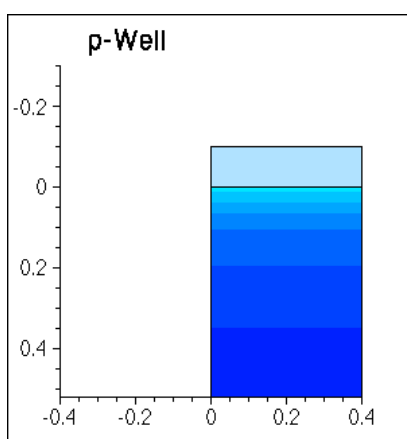


图 4 90nm NMOSFET 的工艺流程 (动画)

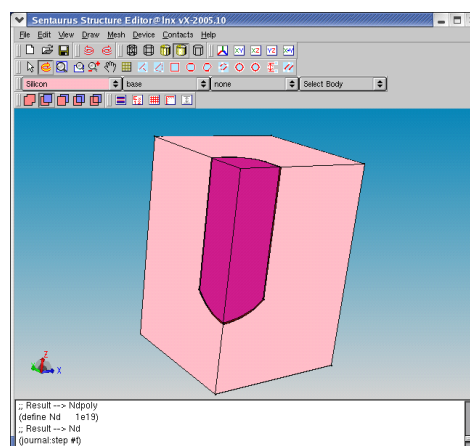


图 5 SDE 的 GUI 界面

B.4 Sentaurus Structure Editor (SDE)

SDE 是一个二维和三维器件编辑器以及三维工艺仿真器。具有三种截然不同的工作模式: 二维结构编辑、三维结构编辑和三维工艺仿真。几何和工艺仿真操作能够自由混合。

利用 GUI, 采用二维或三维元件 (例如: 矩形、多边形、立方体、圆柱体、球体) 以几何形式创建二维和三维器件。也可通过对二维对象进行简单的挤压 (extrusion) 或者使二维对象沿某条路径进行扫描从而获得三维结构。

在工艺仿真模式下 (Procem), SDE 将工艺步骤 (例如刻蚀和淀积) 转换成几何操作。Procem 支持各种选项, 例如各向同性和各向异性刻蚀、淀积、圆化、混合 (blending), 以考虑特殊的工艺效应。

SDE 提供最先进的可视化能力。结构以其创建的方式进行显示 (所见即所得), 功能强大的视图过滤使得可以选择一个子集进行查看或者使特定的区域变得透明。

交互式地定义掺杂分布和网格化策略。布局以半透明的方框进行显示, 以方便确认。支持网格生成工具 Mesh 和 Noffset3D 的所有掺杂和网格化选项。

GUI 还包含一个命令窗口, 在该窗口中显示与 SDE 的 GUI 操作相对应的命令脚本。也可在命令窗口中直接输入脚本命令。

B.5 Mesh and Noffset3D

Synopsys 提供两种最先进的方法用于自动产生网格。**Mesh** 采用基于四叉树/八叉树的方法，产生与坐标轴对齐的网格结构。**Noffset3D** 是 **fully unstructured**，对材料边界处给予了特别的关注。二者均可用于对工艺和器件模拟器的输入进行离散化。（译者注：工艺模拟的过程中一般网格是不断调整的，而器件模拟的过程中网格一般不变。）网格不仅针对几何结构进行自适应调整，而且针对掺杂浓度进行自适应调整以考虑那些高浓度梯度的地方。

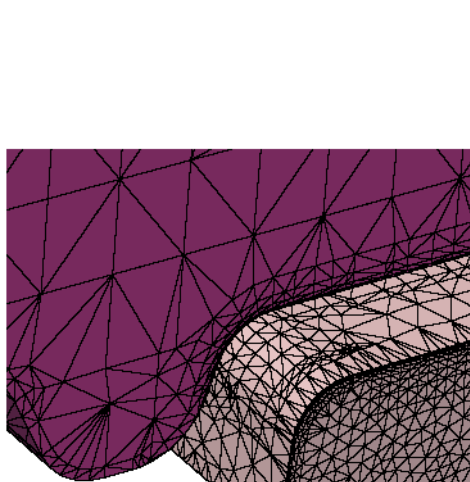


图 6 STI 背面的 Noffset3D 网格。为方便观看，移除了栅氧。图中粉色为沟道区，暗紫色区域为多晶硅栅。

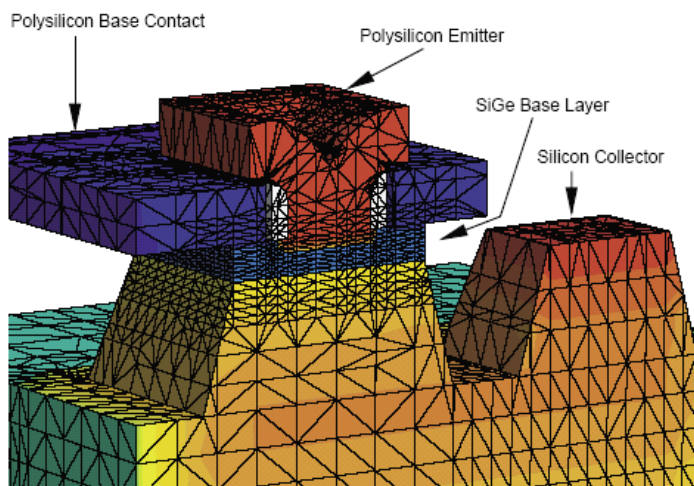


图 7 氮化物边墙（灰色区域）的 SiGe HBT。图中没有给出氧化区域。图中给出了多晶硅和硅区域的掺杂分布。N 掺杂区为红色而 P 掺杂区为蓝色。与坐标轴对齐的网格是由 MESH 生成的。

B.6 Sentaurus Device

SDevice 模拟半导体器件的电、热和光性能。是业界领先的器件模拟器，可以处理一维、二维和三维几何结构以及混合模拟。包含一个复杂的物理模型集合，适用于所有相关的半导体器件和工作条件。

Sentaurus Device 可以在研发的早期用于对器件进行评估、优化、提取 **SPICE 模型** 和统计数据。Sentaurus Device 的应用包括：**VDSM 硅**，Sentaurus Device 被证明在 100nm 以下工艺均相当精确；**SOI 器件**，Sentaurus Device 具有可靠的收敛性和精度；双栅和 FinFET 器件，实现了量子传输；SiGe；薄膜晶体管；光电子；异质结 HEMT 和 HBT；功率和 RF 半导体器件。

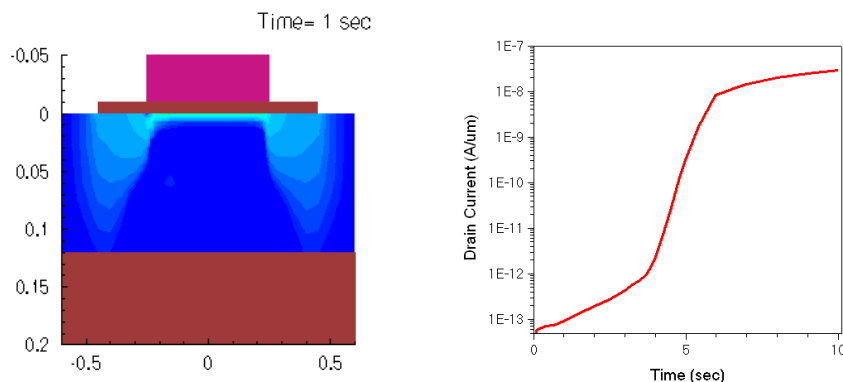


图 8（左）SOI MOSFET 在辐照条件下（剂量率为 50Krad/s）电流密度随时间的变化（动画）。器件的偏置条

件为 $V_{gs}=0V$, $V_{ds}=2V$ 。辐照使陷阱带电, 尤其是在埋氧中 (底部褐色区域)。这将产生一条沿 Si/埋氧界面的漏流通路。电流密度从 $<10^{-5}Acm^{-3}$ (蓝色) 变化到 $>10^3Acm^{-3}$ (红色)。(右) 漏极电流随时间的变化。

B.7 Tecplot SV

Synopsys 集成了 Tecplot (一个用于科学可视化的专用软件), 并对其进行了定制。Tecplot 公司是一家历史悠久的提供高质量工程和科学可视化工具的公司。Tecplot SV 是一个绘图软件, 具有强大的二维和三维功能, 可用于查看模拟和实验数据。Tecplot SV 代表了最先进的科学可视化。除此之外, Tecplot SV 还可用于研究和分析数据, 产生二维和三维视图, 产生具有演讲质量的图表和动画, 以及在 Web 上共享结果。Tecplot SV 的通用性和高质量输出使用户可以获得所需的任何类型的图表。

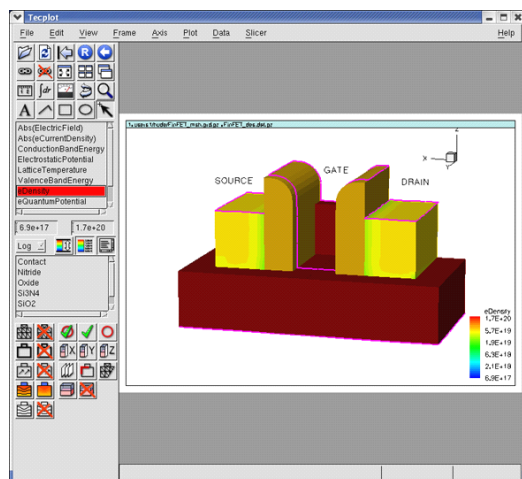


图 9 Tecplot SV 的 GUI, 图中给出了 FinFET 器件的电子密度。

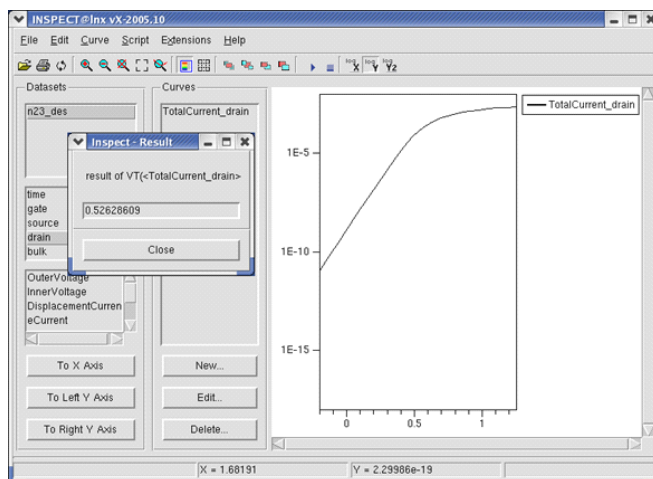


图 10 Inspect 的 GUI, 图中给出了某 I_dV_{gs} 曲线和一个交互式提取阈值电压的对话框。

B.8 Inspect

Inspect 是一个 x-y 数据的绘图和分析工具, 例如半导体器件的掺杂分布和电特性。GUI 使得用户可以快速获得所需的曲线。脚本语言和数学函数库使得用于可以对曲线进行计算, 从曲线中提取所需的数据 (译者注: 例如阈值电压)。提出的数据可以返回 Sentaurus Workbench 用于更深层次的应用。

B.9 Calibration Kit

Calibration Kit 是一个扩展了 Sentaurus Workbench 功能的软件包。Calibration Kit 作为一个数据库浏览器 (包含二次离子质谱 (secondary ion mass spectroscopy, SIMS) 数据的校准库), 一个模拟和工程管理器。Calibration Kit 辅助用户:

- 完成工艺模拟器 (Sentaurus Process, TSUPREM-4, Dios) 的有效校准。
- 获得一个评估工艺条件的快速、精确和可靠的方法。
- 理解工艺对各种控制参数的敏感性。

Calibration Kit 允许用户对工艺设备参数对器件电特性的影响进行一个具有预测价值的分析, 从而使得用户可以高效地优化设备。

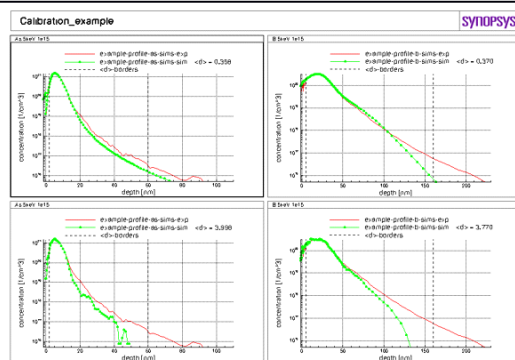


图 11 Tecplot SV 中查看的四个实验。图中比较了注入的实验数据（SIMS，红色曲线），解析模型（上面两幅图中的绿色曲线）和 Monte Carlo 模拟（下面两幅图中的绿色曲线）。（译者注：从图中可以看出，**Monte Carlo 模拟并不一定比解析模型好**）

SWB

SWB 是一个可视化的集成环境，用于设计、组织和运行模拟。

A. Getting Started（15 分钟）

目标：SWB 概述。

A.1 概述

特定的模拟以工程的形式进行组织。SWB 自动管理信息流，包括用户输入文件的前处理、参数化工程、设置和执行特定的工具以及查看结果。

SWB 允许用户定义参数和变量以进行复杂的参变量分析。结果可输入到统计软件和电子表格软件。

A.2 启动 SWB

在启动 SWB 之前，用户必须设置 STDB 环境变量。该变量被设置为一个目录路径，通常为用户 HOME 目录下的 DB 或 STDB 目录。

例如：

1、在用户 HOME 目录下

`B2500-4%mkdir STDB`

2、编辑.cshrc

`setenv STDB $HOME/STDB`

然后，启动 SWB

`B2500-4%swb &`

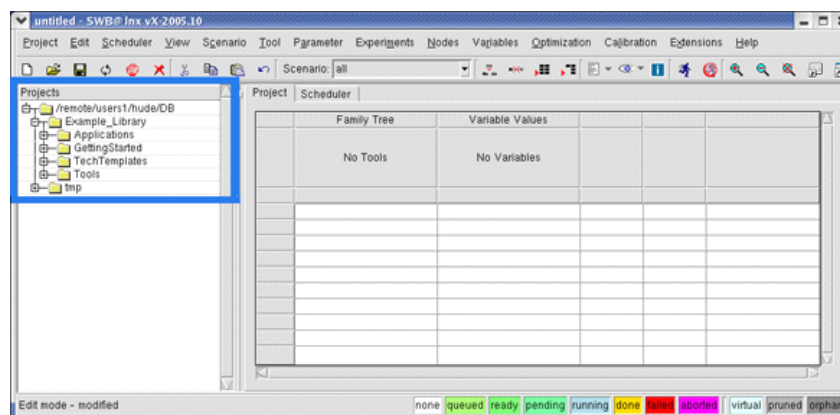



图 1 SWB 主窗口

SWB 启动之后，将会搜寻 STDB 环境变量，在工程窗口中列出其内容（见图 1 蓝色框所示区域）。如果是有效的 SWB 工程，则在文件夹或者工程名前有一个图标 .

B. 运行工程（30 分钟）

B.1 加载 SWB 工程

从 Examples Library 中 copy 一个工程，如：

```
B2500-4%cd $STDB
```

```
B2500-4%cp -r /CAD/Synopsys/sentaurus2006.06/tcad/Y-2006.06/Example_Library/GettingStarted/SWB_nmos ./
```

启动 swb，然后双击工程窗口中的 SWB_nmos，打开 SWB_nmos 工程（见图 1）

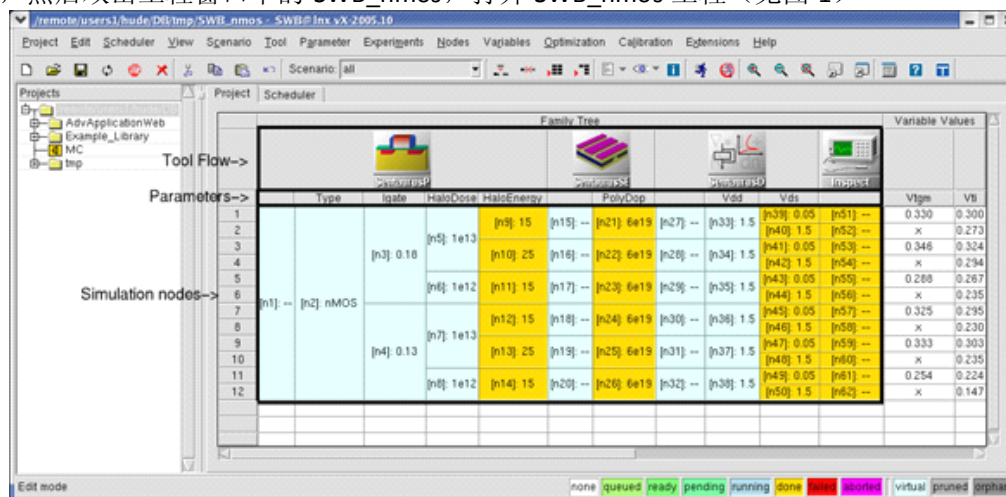


图 1 swb 的主窗口，图中显示了工具流、参数和模拟节点（黑色边框）

在 SWB_nmos 例子中，工具流包括 Sprocess, SDE, Sdevice 和 Inspect。在工具流下面，列出了[n1],[n2]等模拟节点。

（注：如果模拟节点数字没有显示，可以 View->Tree Options->Show Node Numbers）

模拟节点的一个完整序列形成一个实验（experiment）。如果采用参数的话，一个工具流可以对应任意多个实验。

在工具流的右边，给出了变量和从 Id-Vd 曲线中提取的电参数，包括：Vtgm, Vti, Id, SS, gm, Lgeff（有效栅长），Xj（结深），Ygox, Tox（栅氧厚度）。模拟结束后，提取的电参数显示在对应的列中。

双击节点可以显示节点的属性（见图 2）。

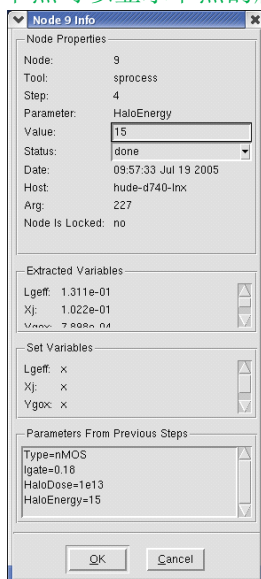


图 2 节点 9 的节点信息对话框

在节点属性中，显示了节点序号，对应的工具，相应的参数及其值，节点状态，模拟的日期和时间，模拟运行所使用的计算机，模拟耗费的总时间。在所有这些信息中，只有节点状态是可以改变的，在很多情况下这是一个有用的特性。

节点信息对话框也显示了提取的和设置的变量（Lgeff,Xj,Ygox,Tox），以及前面步骤所用的参数及其值。

B.2 改变树的显示属性

工程中的每个模拟节点都有一种颜色显示其状态。右下角的颜色表显示了每种颜色代表的含义。

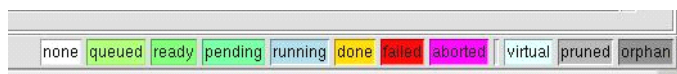


图 3 颜色表示不同的节点状态

例如，当工程 SWB_nmos 打开时，节点为黄色，表明节点已经被模拟过，蓝色给出工艺和器件模拟步骤的一些信息。这是因为 SWB_nmos 工程是 copy 过来的，并且是成功运行过的。为了确认，可以双击节点并查看其状态。（将状态从“done”改为“aborted”或者“running”，观察节点颜色的改变。）

SWB 工程显示的样式是非常灵活的，用户可以调整。用户可以只显示工具流，或者为不同的模拟节点排序，或者显示参数、变量和提取的值以及其它信息。更多的信息，请参考：

View->Tree Options

B.3 清空工程目录

在运行 SWB_nmos 工程之前，清空前次运行获得的工程结果。

- 1、Project->Clean Up
- 2、在弹出的对话框中，选择需要移除的项（见图 4）
- 3、单击 OK

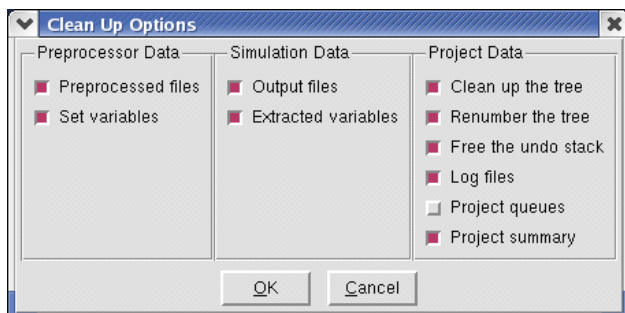


图 4 清空选项对话框

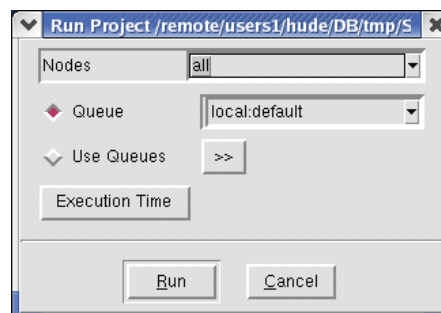


图 5 运行工程对话框

SWB 删除所有与前次运行相关的文件，准备进行全新的运行。在图中节点的颜色会发生相应的改变，从黄色（done）变为白色（White）。

B.4 运行工程

- 1、在 SWB_nmos 上点击右键，选 Project->Run（见图 5）

如果只需要运行一部分节点，在 Nodes 域中选择需要运行的节点的序号。例如，如果只需运行 Sentaurus Structure Editor 的前两个节点（节点 15 和 16），在节点域中输入 15 16，单击 Run。

在运行一个模拟节点之前，SWB 对输入文件进行预处理，亦即：检查语法错误，将参数和变量用相应的值替换。在成功进行预处理之后，SWB 以下面的形式创建模拟输入文件：

pp<node_number>_<tool_name>.cmd 和 pp<node_number>_des.par（Sdevice 参数文件）

只有当这些文件创建之后，节点的模拟才会开始。预处理可以在节点的模拟开始之前自动进行，也可由用户在不进行模拟的情况下单独执行预处理。

只进行预处理：

在 SWB_nmos 上点击右键，选 **Project->Preprocess**
预处理成功完成后，SWB 在 Project Log 对话框中相应的信息。
此时，所有变量（例如 Vtgm、Vti 和 Id）被提取（采用 Inspect，节点 51-62），并显示在工具流的右边。

B.5 查看输出结果

每个特定的节点都有一些输入和输出文件与之关联。这可以通过右键点击一个节点并选择 **Visualize** 查看。可以采用 SEdit 查看所有的文本和日志文件。
所有的输出数据文件（包括 Dios、Sprocess、Mdraw、Sdevice、Sdevice EMW 的输出）均可采用 Tecplot SV 查看，所有的曲线（Dios 和 Sdevice 输出的.plx 和.plt 文件）均可采用 Inspect 查看。
除此之外，运行模拟的过程中写到标准输出的信息，可以通过选择 **View Output** 选项查看。

C. 创建工程（30 分钟）

目标：装配和保存一个新的 SWB 工程。

C.1 设置工具流

设置工具流的第一步是创建一个工程：

Project->New

这将在 tmp 文件夹中创建一个临时工程，工程名类似 g_lnx_2879_0.tmp（见图 1）。

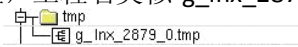


图 1 工程目录

该工程需要用到的工具包括 Sprocess, SDE, Sdevice 和 Inspect。

设置一个工具流：

在 No Tools 处点击右键（见图 2）

	Family Tree	Variable Values
	No Tools	No Variables

图 2 显示了 Family Tree 的主窗口

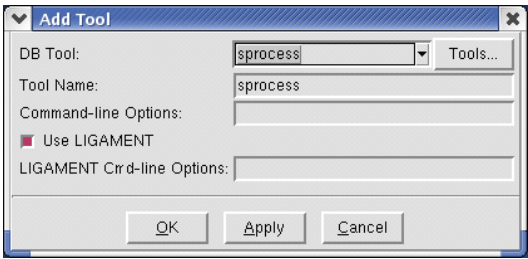


图 3 添加工具对话框

选择 **Add**，弹出如图 3 的对话框。

点击 OK，出现一个 Create Default Experiment 对话框，在该对话框中，可以创建一个 scenario（见图 4），在 Scenario 域中输入 new。

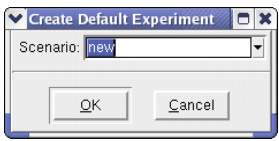


图 4 Create Default Experiment 对话框

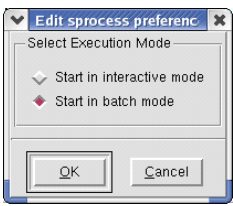


图 5 Sprocess 的喜好选择对话框

设置 Sprocess 以批处理方式运行：

右键点击 Sprocess 图标，Edit Input->Preferences

现在，必须提供 Ligament 和 Sprocess 使用的命令和版图文件。由于工程与 SWB_nmos 完全相同，可以从 SWB_nmos 工程中导入所需的文件。

- 1、右键点击 Sprocess 图标，选择 Import File->Ligament Flow。
- 2、在 Import Flow File 对话框中选择 sprocess_lig.cmd（见图 6）

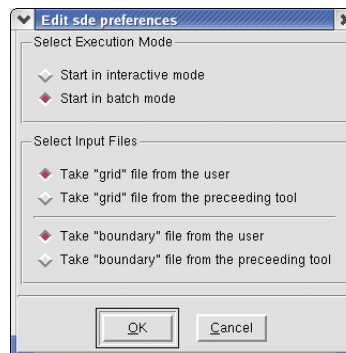
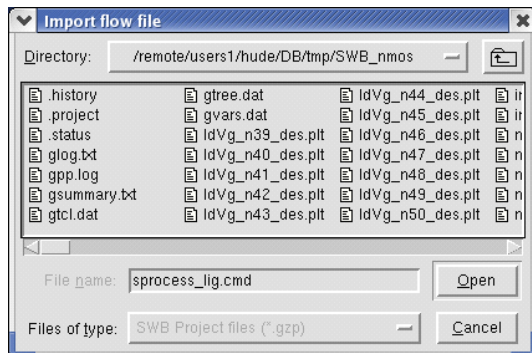


图 6 Import Flow File 对话框 图 7 SDE 喜好选择对话框

类似地，右键点击 Sprocess 图标，选择 Import File->Ligament Layout，选择 sprocess_lig.lyt 文件。

添加 SDE 工具（见图 7，注：选择批处理运行方式；不选 Use Ligament。）

现在，必须提供命令文件和边界文件。

右键点击 SDE 图标，Import File->Commands，选择 sde_dvs.cmd。

添加 Sdevice，导入 sdevice_des.cmd 文件。类似地，Import File->Parameter(Si Model)，选择 sdevice.par 文件。

添加 Inspect，导入 inspect_ins.cmd 文件。

译者注：

- 1、各种工具的输入文件其实是以 Import File 的方式导入，否则，工具不知道哪个文件是其输入文件；
- 2、Q：这一个工具的输出作为下一个工具的输入，SWB 是以一种什么样的方式传递信息？基于默认的文件名？
- 3、为了规范化，推荐使用固定的文件名
- 4、可以将带参数的命令文件理解为模板文件，在参数被实际值替换之后的文件为实际运行的文件

D. 组装多个实验（30 分钟）

目标：添加参数和变量，创建多个实验和方案。

D.1 添加参数

在 SWB 中，可以定义参数并赋予参数多个值从而创建实验分支。参数的每个值创建一个实验，因此，如果一个参数有 N 个值，则对应 N 个实验。如果参数 P1 和 P2 分别对应 M 和 N 个值，则对应 M*N 个可能的实验。

下面介绍 Sprocess, SDE 和 Sdevice 中的参数。（注：各工具的命令文件均可引用 SWB 设置的参数。）

在 Sprocess 中，具有下列参数：

- Type 用于定义结构名称。
- lgate 用于定义栅长。
- HaloDose 用于定义 Halo 注入的剂量。

- HaloEnergy 用于定义 Halo 注入的能量。

在 SDE 中，PolyDop 用于指定多晶硅区域的掺杂浓度。

在 Sdevice 中，具有下列参数：

- Vdd 定义施加的工作电压。
- Vds 用于定义漏极电压。

添加参数：

右键点击 Sprocess 图标下的灰色框，选择 Add（见图 1）

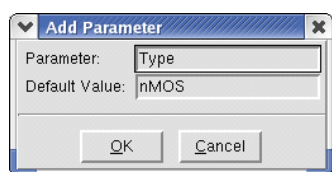


图 1 添加参数对话框

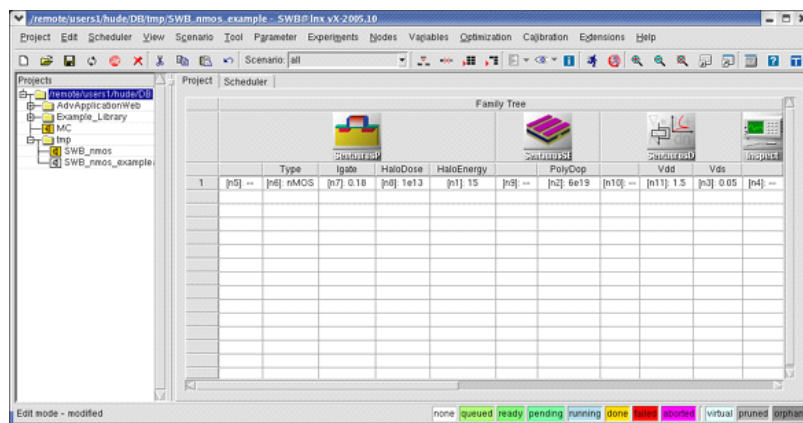


图 2 SWB 的主窗口，图中显示了新的参数

类似地添加其它参数。在添加完各种参数之后，主窗口应当如图 2 所示。

为了检查参数是否已插入到 SDE 的输入文件 sde_dvs.cmd 中：

右键点击 SDE 图标，选择 Edit Input->Commands 打开 sde_dvs.cmd

(define PolyDop @PolyDop@)

对于 Sdevice，Vdd 和 Vds 以如下形式插入：

{ Name="gate" Voltage=!(puts [expr \$SIGN*@Vdd@])! }

{ Name="drain" Voltage=!(puts [expr \$SIGN*@Vds@])! }

@PolyDop@, @Vdd@和@Vds@表明它们是参数，在预处理的过程中将被参数定义中的实际值替换。本工程中引入的所有参数值可以在预处理文件中查看。

对工程进行前处理：

右键点击工程图标，选择 Project > Preprocess

D.2 设置多个实验

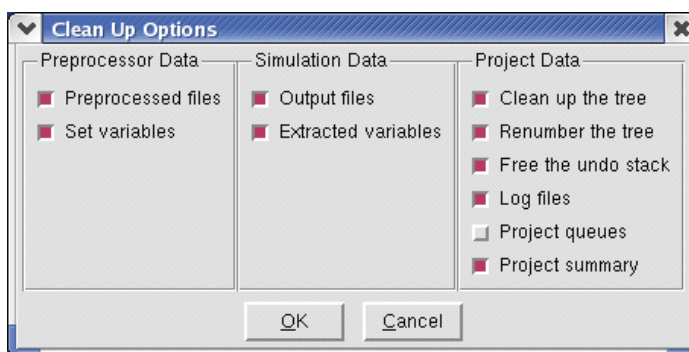
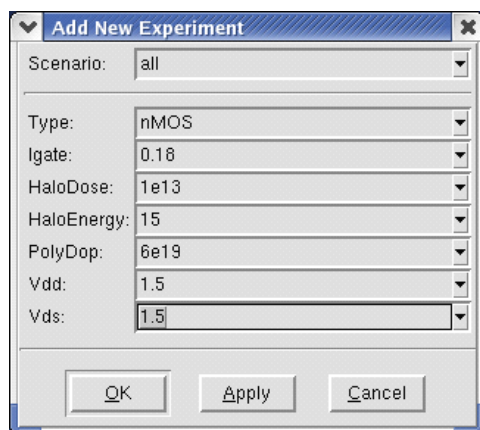


图 4 创建新的实验

图 5 清空工程

创建多个实验：

- 1、Experiments->Add New Experiment
- 2、输入图 4 中所示的值

采用以下值重复以上过程：

Experiment 3	Experiment 4	Experiment 5	Experiment 6	Experiment 7
Type : nMOS	Type : nMOS	Type : nMOS	Type : nMOS	Type : nMOS
Igate : 0.18	Igate : 0.18	Igate : 0.18	Igate : 0.18	Igate : 0.13
HaloDose : 1e13	HaloDose : 1e13	HaloDose : 1e12	HaloDose : 1e12	HaloDose : 1e13
HaloEnergy : 25	HaloEnergy : 25	HaloEnergy : 15	HaloEnergy : 15	HaloEnergy : 15
PolyDop : 6e19	PolyDop : 6e19	PolyDop : 6e19	PolyDop : 6e19	PolyDop : 6e19
Vdd : 1.5	Vdd : 1.5	Vdd : 1.5	Vdd : 1.5	Vdd : 1.5
Vds : 0.05	Vds : 1.5	Vds : 0.05	Vds : 1.5	Vds : 0.05
Experiment 8	Experiment 9	Experiment 10	Experiment 11	Experiment 12
Type : nMOS	Type : nMOS	Type : nMOS	Type : nMOS	Type : nMOS
Igate : 0.13	Igate : 0.13	Igate : 0.13	Igate : 0.13	Igate : 0.13
HaloDose : 1e13	HaloDose : 1e13	HaloDose : 1e13	HaloDose : 1e12	HaloDose : 1e12
HaloEnergy : 15	HaloEnergy : 25	HaloEnergy : 25	HaloEnergy : 15	HaloEnergy : 15
PolyDop : 6e19	PolyDop : 6e19	PolyDop : 6e19	PolyDop : 6e19	PolyDop : 6e19
Vdd : 1.5	Vdd : 1.5	Vdd : 1.5	Vdd : 1.5	Vdd : 1.5
Vds : 1.5	Vds : 0.05	Vds : 1.5	Vds : 0.05	Vds : 1.5

将工程清空（见图 5）：

此时，主窗口应当如图 6 所示：

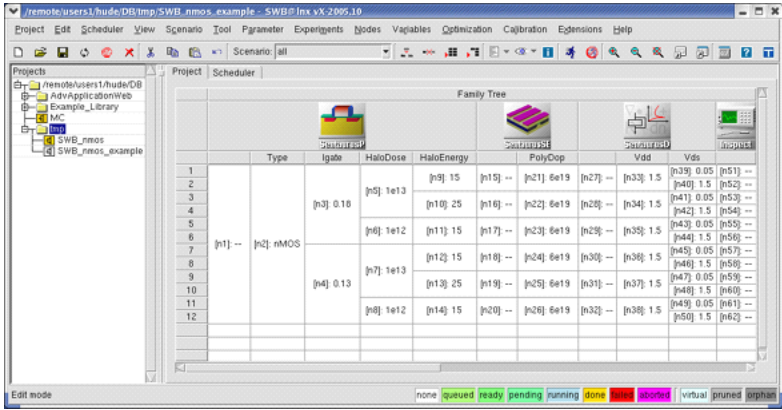


图 6

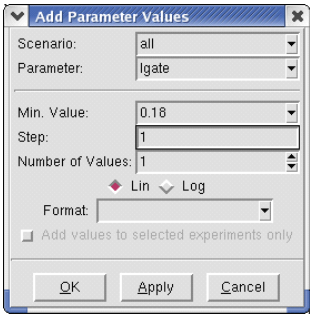


图 7

注：如需渐进地改变某个值，右键点击参数名，选择 Add Values（见图 7）

D.4 创建方案

通过选择不同的参数值，共有 12 个实验。用户可以运行所有的 12 个实验（这一方案被称为 all），也可运行部分实验（称为一个方案）。

除此之外，用户可以复制实验子集以创建多个方案，然后单独运行各个方案。下面举例说明，创建一个 New1 方案，包括实验 3，7 和 9。

- 1、单击实验 3 加亮。
- 2、Ctrl+单击，加亮实验 7 和 9。

3、Edit->Copy。

4、Scenario->Add。

5、在弹出的对话框中，输入 New1，点击 Add（见图 8）。

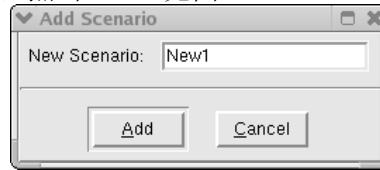


图 8 添加方案

这将创建一个名为 New1 的方案，实验数为 0。现在，Edit->Paste。上面所选的 3 个实验被 copy 到该方案，该方案可以以通常的方式运行。

通过滚动工具栏中的 Scenario 菜单（图 9 中黑框所示）可以在 SWB 主窗口中显示不同的方案。

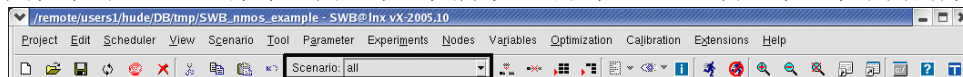


图 9 工具栏中的 Scenario 菜单

D.5 裁剪工程树

当工程中具有许多参数时，经常出现某些参数组合不需要模拟的情况。此时，可以通过去除这些实验以裁剪工程树。

例如，假设方案 New1 中 $I_{gate}=0.18$, $HaloDose=1e13$, $HaloEnergy=25$ 和 $V_{ds}=0.05$ 的情况不需要模拟。因此，可以裁剪掉节点 41：

1、选择节点 41

2、Node->Prune

节点 41 和 53 被裁剪掉并以灰色显示。

D.6 添加变量

在许多情况下，参数可以用变量替换以缩短模拟树。这些变量包含和引用的方式与参数类似。下面举例说明，定义一个 V_{sub} 变量代表衬底偏压。

Ligament

Ligament 是一个高级工艺流程编辑器，为用户提供一个十分方便的 GUI 用于创建工艺模拟器的输入文件。

A. Getting Started 10 min

A.1 简介

Ligament 的输入包括工艺流程、工艺库、版图以及与 TCAD 相关的信息。输出是目标工艺模拟器的命令文件。Ligament 包括三个工具：流程编辑器、版图编辑器和转换器。

A.2 Ligament 流程编辑器

在流程编辑器中，用户采用宏组织工艺流程，包括用户自定义的局部宏和 Ligament 库中定义的全局宏。

A.3 Ligament 版图编辑器

版图编辑器的主要目的是提供一个 EDA 版图工具和 TCAD 之间的接口，例如定义一维/二维/三维模拟区域、参数化版图（例如：Vt roll-off 模拟）。版图编辑器支持 GDSII 和 CIF 格式的版图。

A.4 Ligament Translator

将 SPR 语言转换为目标模拟器的语言。转换器支持的工艺模拟器包括 Sprocess、TSUPREM-4、Dios 和 3D 工艺仿真器 SDE。

B. Ligament 流程编辑器 30 min

B.1 启动 Ligament 流程编辑器

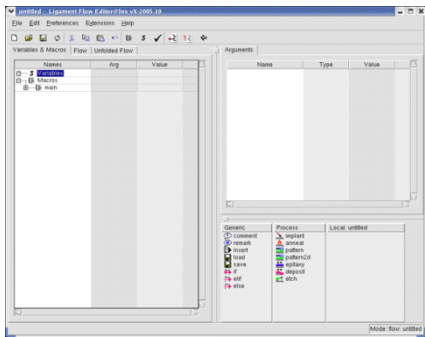


图 1 流程编辑器主窗口

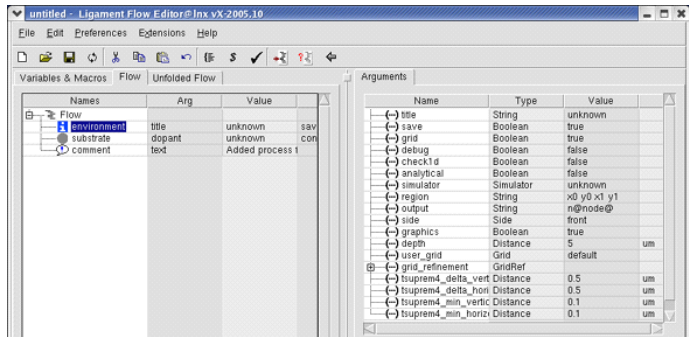


图 2 带宏调用的工艺流程

%ligedit

显示如图 1 所示的主窗口。

B.2 开始装配一个新的工艺流程

开始一个新的工艺流程

- File > New

设置流程的初始环境：

- Edit > Add Process Header

三个新的宏被插入到工艺流程的开头（见图 2）

B.4 改变宏调用的参数

environment		
Argument	Value	Comment
title	A simple process flow	Title of process simulation
simulator	sprocess	Target simulator
region	0 0 1 0	Simulation domain
output	simple	File name of the translated target

substrate		
Argument	Value	Comment
dopant	boron	Substrate dopant
concentration	5e14 /cm3	Dopant concentration

comment		
---------	--	--

Argument	Value	Comment
text	A simple process flow for sprocess simulation to show the use of Ligament Flow Editor.	

B.7 用户定义的变量

定义一个新的变量：

- File > New Variable
- 将其名称改为 **damage**，敲回车。
- 在 **Argument** 选项卡内双极该变量名。
- 将变量的类型改为 **Type**，值改为 **defect.model=plus.one**（见图 9）

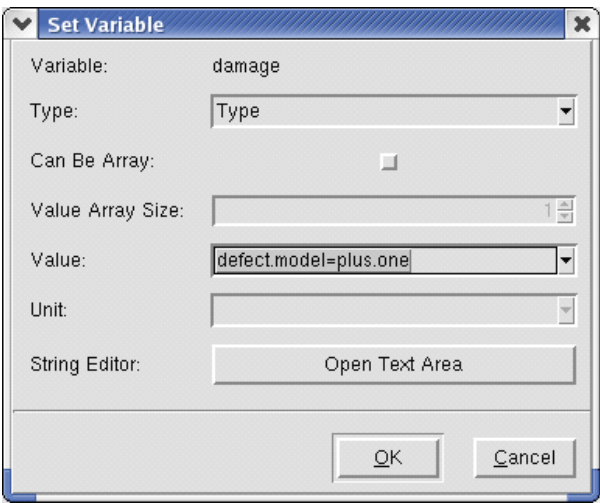


图 9

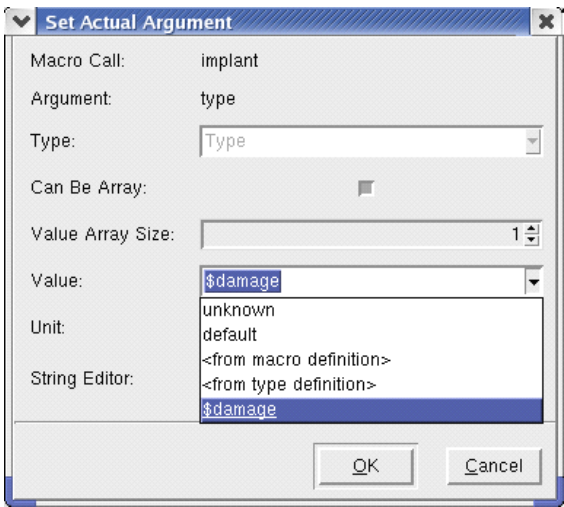


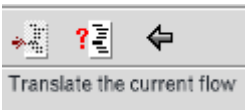
图 10

使用变量：

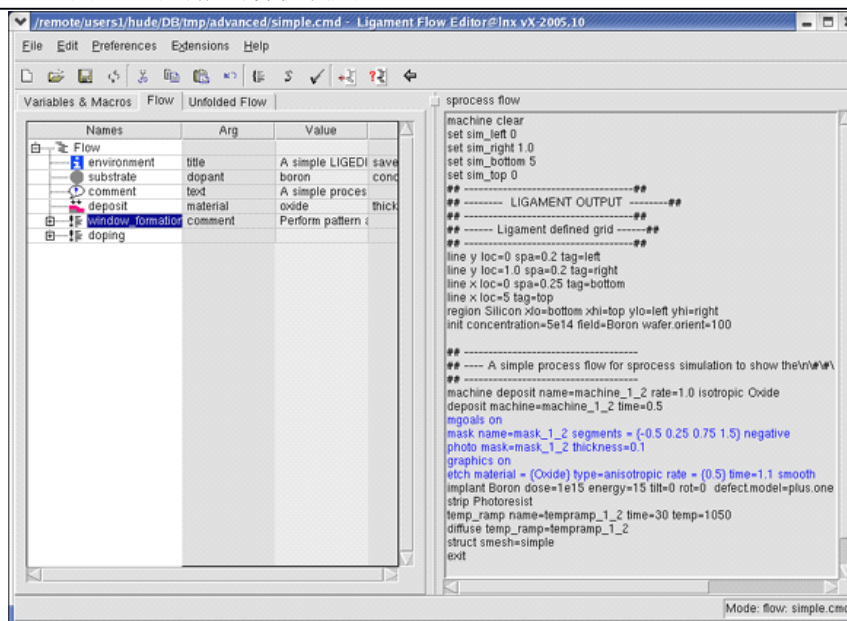
- 选择 **implant** 宏调用
- 双极 **type** 参数
- 将值设为 **\$damage**（见图 10）

注：通常，在 **Ligament** 流程编辑器中可以访问工艺步骤的绝大部分重要参数。但是，**Ligament** 流程编辑器不可能包括特定模拟器的所有命令和选项。为了克服这一问题，**Ligament** 流程编辑器采用通用宏 **insert**，可以向工艺流程中插入特定模拟器的命令。另一方面，与模拟器相关的命令选项（如 **damage.model=plus.one**），通常使用 **Type** 变量解决（如本节所示）。

B.8 转换和语法检查



以上三个工具栏命令分别为：转换、检查语法、回到编辑模式。
转换之后得到的命令文件显示在右边，注意：可以交叉显示（见下图）。



C. Ligament 版图编辑器 30 min

C.1 启动 Ligament 版图编辑器

%prolyt

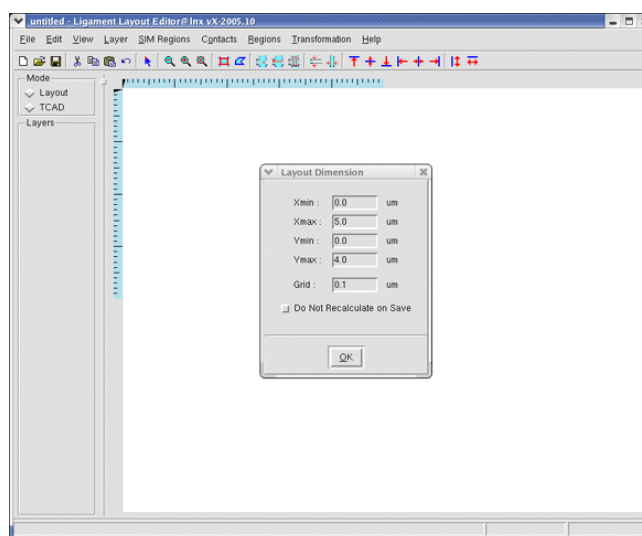


图 1 版图编辑器主窗口

注：Layout 模式使能典型的版图编辑功能，TCAD 模式允许用户访问 TCAD 的特殊功能。

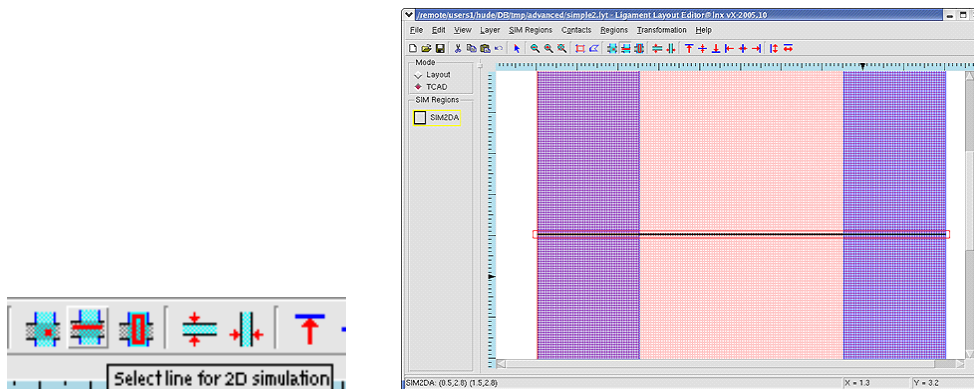
C.4 保存版图

File > Save Layout As > DF-ISE

版图以.lyt 扩展名保存。保存的版图文件为 DF-ISE 数据格式，包含版图的几何结构信息以及定义的模拟区域。

C.6 定义模拟区域

1. 选择 **TCAD** 模式
2. 单击"**select line for 2D simulation**"图标

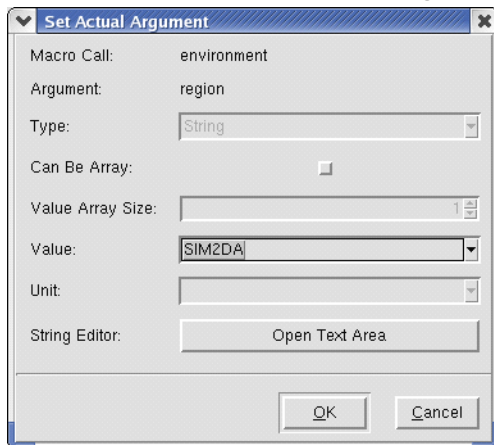


3. 画一条割线

C.7 使用版图

Ligament 版图编辑器中定义的 mask 和模拟区域，能够在 Ligament 流程编辑器中使用。
在流程编辑器中：

- 加载版图：File > Open Layout
- 选择 environment 宏，双击 region 参数：



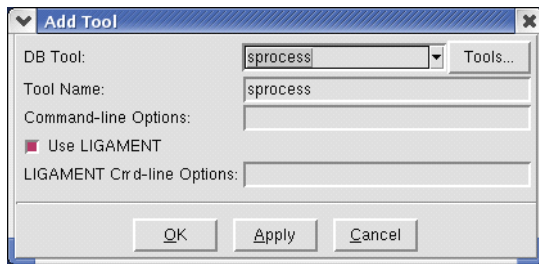
注：SIM2DA 是在版图编辑器中定义模拟区域时指定的名称。

D. 在 SWB 中运行 Ligament 30 min

D.1 在 SWB 中启动 Ligament 工具

在工程中添加 sprocess 工具：

Tools > Add



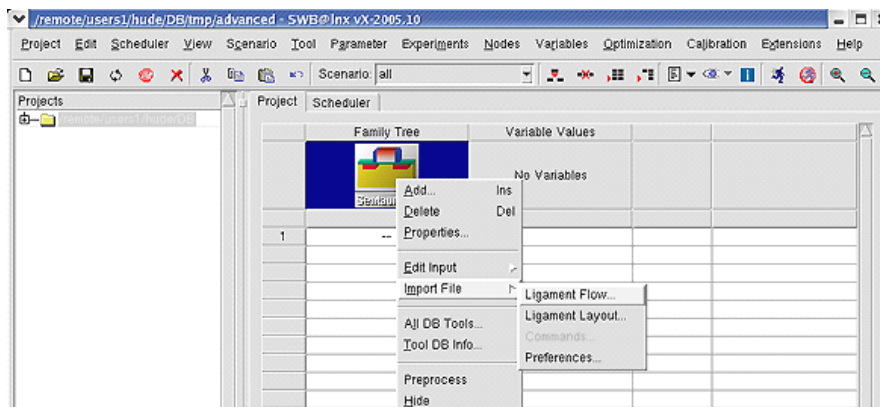
启动 Ligament 工具:

右击 sprocess 图标, **Edit Input > Ligament Flow**

右击 sprocess 图标, **Edit Input > Ligament Layout**

D.2 导入 Ligament 文件

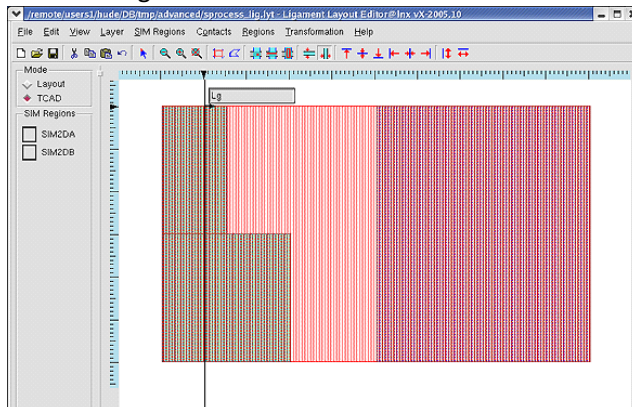
Import File > Ligament Flow (见下图)



D.5 Ligament 版图编辑器中的版图参数化

在版图编辑器中:

1. 选择 TCAD 模式
2. 单击 Create a vertical layout parameter 图标
3. 将鼠标移到 $x=0.2$ 的位置, 单击
4. 输入 Lg 作为参数名



保存版图参数定义:

1. File > Save Layout Command As

2. 输入 sprocess_lig.lcf 作为文件名

D.6 SWB 中的参数化

在版图编辑其中定义了参数之后，在 SWB 中也必须定义相应的参数。

在 SWB 进行预处理时，版图参数由 SWB 中定义的值替换，从而实现了版图参数化。

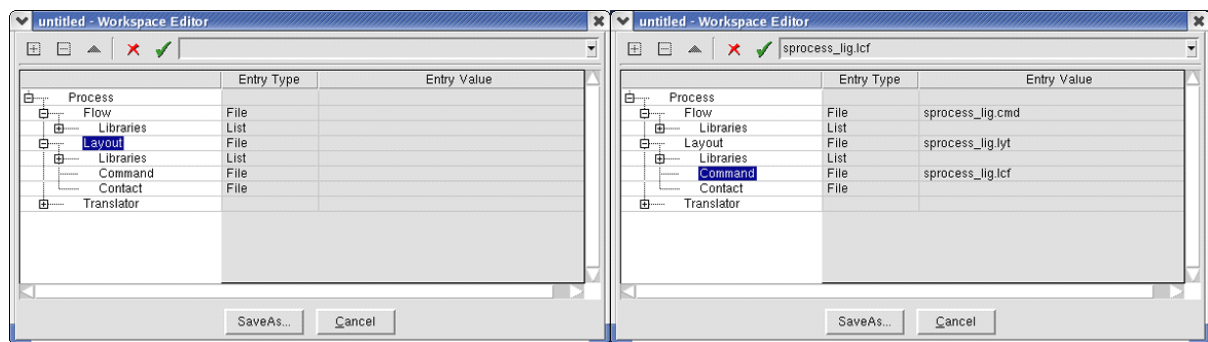
D.7 Ligament 工作区

Ligament 工作区包含到工艺流程文件、版图文件和命令文件的引用。除此之外，Ligament 工具的命令行选项也可在工作区文件中指定。

使用 Ligament 工作区简化了单个 Ligament 工具的使用。

建立一个 Ligament 工作区：

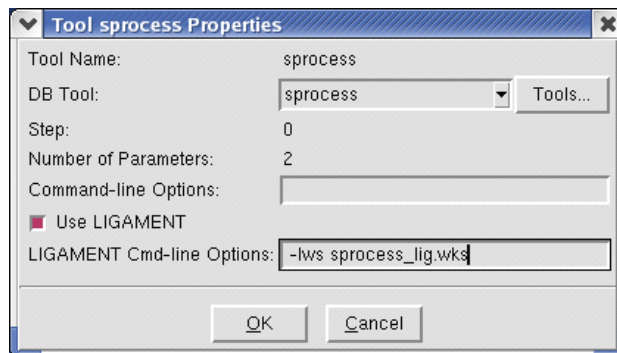
- File > Workspace > New



- 保存工作区文件.wks。

在 SWB 中使用 Ligament 工作区：

右击 Sprocess 图标，选择 Properties



D.8 SWB 的预处理和 Ligament 转换

本例中的 Ligament 工作区文件 sprocess_lig.wks 包含工艺流程文件 sprocess_lig.cmd，版图文件 sprocess_lig.lyt 和版图命令文件 sprocess_lig.lcf，这些文件使用的都是 SPR 语言，需要转换到特定工艺模拟器对应的语言。

在预处理时，SWB 自动执行转换过程。

转换过程包括两个步骤：

第一步，调用 Ligament 转换器，将工艺流程展开（unfold），还有可能对工艺流程进行分割（split）。

unfold

```
%ligament -unfold sprocess_lig.cmd
```

在本例中，由于在 Tool Properties 对话框中定义了 Ligament 的命令行选项，实际执行的命令为：

```
%ligament -unfold sprocess_lig.cmd -lws sprocess_lig.wks
```

另外，这一步还执行参数替换（使用 SWB 中定义的值）。这一步的输入包括 sprocess_lig.cmd、sprocess_lig.lyt 和 SWB 中定义的参数。

在这一步中，所有的 SWB 参数都用节点的参数值进行了替换。但是，Ligament 命令的定义、掩模的定义以及版图参数化还没有解决。

这一步产生的文件为 pp3_lig.cmd 和 pp4_lig.cmd。

第二步，调用 Ligament 转换器，对前一步产生的 pp3_lig.cmd 和 pp4_lig.cmd 进行处理，生成最终的工艺模拟命令文件 n3_fps.cmd 和

n4_fps.cmd。

```
%ligament -lws sprocess_lig.wks -flow pp3_lig.cmd -layoutcmd pp3_lig.lcf
```

在这一步中，所有的 Ligament 命令、版图掩模以及版图参数化被处理。

Sentaurus Process

SProcess 是一个完整的、高度灵活的多维工艺建模环境。

A. Getting Started 15min

目标：

- 对 Sprocess 的特征做一个简介
- 搞清楚 Sprocess 使用的文件类型
- 搞清楚如何运行 Sprocess

A.1 简介

Sprocess 采用验证过的校准方法与最新的实验数据进行过校准，能够提供具有预测价值（predictive，也就是说精度很高）的结果。

通过一系列复杂性不断增加的例子，介绍 Sprocess 的绝大部分广泛使用的命令。

A.2 Sprocess 使用的文件类型

- Sprocess 命令文件(*.cmd)
- 日志文件(*.log)
- TDR 边界文件(*_bnd.tdr)

通常，用于在工艺模拟结束时保存一个边界文件存储器件的几何结构信息。这个文件可以作为网格化引擎(Mesh 和 Noffset2D/3D)的输入文件。可用 Tecplot 查看该文件。还可以用 TDR 文件查看器查看该文件(%tdx xx.tdr)。

注：也可保存 DF-ISE 格式的边界文件*_mdr.bnd。该文件也可作为网格化引擎的输入。

- TDR 网格和掺杂文件(*_fps.tdr)

该文件包含两种信息：器件的几何结构和网格；器件的掺杂分布。可以采用 tecplot 查看。

注：如果采用 DF-ISE 格式，则网格文件和掺杂分布文件分别以.grd 和.dat 的格式保存。

- DF-ISE 掺杂和网格优化（refinement）文件(*_msh.cmd)

该文件保存掺杂和网格优化信息，与边界文件一起作为网格化引擎的输入。用户通常在模拟结束时保存一个_msh.cmd 文件。

Q：为什么在做 Sprocess 模拟时有时没有生成_msh.cmd 文件？

A：如果在 struct 命令中使用 ise.mdraw，则会生成_mdr.cmd 文件。此文件可以作为网格化引擎的输入。

- DF-ISE 文件(*.plx)

保存某截面掺杂浓度或其它信息的一维分布，可作为 Inspect 的输入。

A.3 启动 Sprocess

```
%sprocess
```

B. 一维工艺模拟 45min

B.1 简介

B.2 定义初始一维网格

```
line x location=0.0      spacing= 1<nm> tag=SiTop
line x location= 10<nm> spacing= 2<nm>
line x location=2.0<um> spacing=0.2<um> tag=SiBottom
```

在绝大部分情况下，长度单位的默认值为 um。显示指定单位时需要采用<>。

B.3 定义初始模拟域(domain)

```
region silicon xlo=SiTop xhi=SiBottom
```

通常，初始的模拟域(domain)可以包含多个区域(region)。

B.4 初始化

```
init concentration=1.0e15<cm-3> field=Boron
```

B.5 设置 MGOALS 网格化策略

在遇到第一个改变几何结构的命令(例如：氧化、淀积、刻蚀)之前，均采用初始网格。对于氧化、淀积、刻蚀这些步骤，必须指定网格重定义策略。Sprocess 采用 MGOALS 作为网格重定义引擎。（注：MGOALS

也执行刻蚀和淀积）

MGOALS 尽量保留初始网格，只修改新层中的网格和新界面附近的网格。

```
mgoals on min.normal.size=3<nm> max.lateral.size=0.2<um> \
    normal.growth.ratio=1.4
```

其中：

- min.normal.size 定义从界面算起，第一层的网格间距。
- max.lateral.size 限制二维模拟时的横向网格间距以及任何位置的最大网格间距。
- normal.growth.ratio 定义从一层到下一层时网格间距的最大变化速率。

B.6 Growing Screening Oxide

```
gas_flow name=O2_1_N2_1 pressure=1<atm> \
    flowO2=1.2<l/min> flowN2=1.0<l/min>
diffuse temperature=900<C> time=40<min> gas_flow=O2_1_N2_1
grid remesh
```

grid remesh 强制执行一个 MGOALS 网格重定义步骤。推荐在氧化步骤完成之后执行此命令。刻蚀和淀积采用 MGOALS 执行，自动执行网格重定义。

B.7 测量氧化物的厚度

select z=Boron 选择 B 的浓度进行后处理。

layers 打印一个区域的列表，包括各个区域的 top 和 bottom 坐标，以及被选择的实体（quantity）在每个区域内的积分。例如，此处打印了每个区域内 B 的积分浓度。

```
{      Top      Bottom      Integral      Material}
{-6.43e-03  4.11e-03  1.54e+09      Oxide}
{ 4.11e-03  2.00e+00  1.98e+11      Silicon}
```

此处，在热氧化的过程中消耗了 4.11nm 的 Si，最终的氧化物厚度为 $4.11+6.43=10.54\text{nm}$ 。Si 层中的积分浓度为 $1.98 \times 10^{11} \text{cm}^{-2} = 10^{15} \text{cm}^{-3} (2 \times 10^{-4} \text{cm} - 4.11 \times 10^{-7} \text{cm})$ 。

B.8 Depositing Screening Oxide

除了通过热氧化的方法生长氧化物之外，还有另一种模拟速度更快的方法：首先淀积一层氧化物，然后进行热扩散。如果氧化物增强的扩散（oxidation-enhanced diffusion,OED）和氧化过程中的 Si 消耗不重要的话，可以采用这种方法。（译者注：场氧的生长不推荐使用这种方法）

```
deposit Oxide type=isotropic thickness=10.0<nm>
diffuse temperature=900<C> time=40<min>
```

B.9 Tcl 控制语句

在 Sprocess 命令文件中可以使用 Tcl 语法，如：

```
set SCREEN Grow
if { $SCREEN == "Grow" } {
```

```
#--- Growing screening oxide -----
gas_flow name=O2_1_N2_1 pressure=1<atm> \
      flowO2=1.2<l/min> flowN2=1.0<l/min>
diffuse temperature=900<C> time=40<min> gas_flow=O2_1_N2_1
grid remesh

} else {
#--- Depositing screening oxide -----
deposit Oxide type=isotropic thickness=10.0<nm>
diffuse temperature=900<C> time=40<min>
}
```

B.10 注入

```
implant Arsenic energy=50<keV> dose=1e14<cm-2>
      tilt=7<degree> rotation=0<degree>
```

B.11 Saving the as-implanted Profile

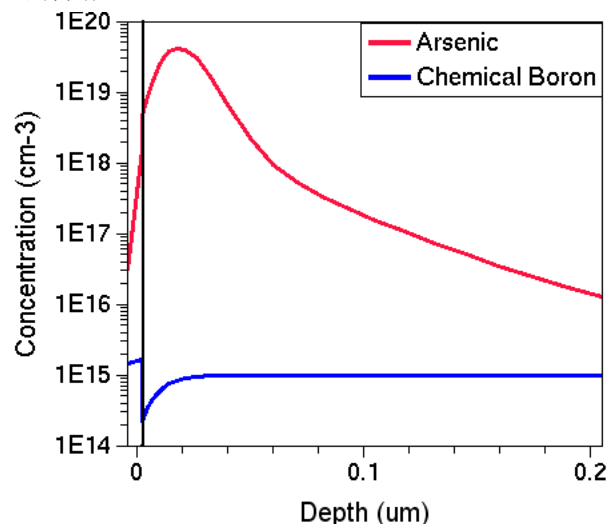
```
SetPlxList { BTotal Arsenic_Implant }
```

```
WritePlx 1DasImpl.plx
```

SetPlxList 命令指定在 PLX 文件中保存哪些变量。如果没有 SetPlxList 命令，则保存所有的变量。

如果是二维和三维结构，在 WritePlx 命令中需要指定一维割线的 x、y 或者 z 坐标。

图 2 给出了 Ar 的掺杂分布以及 B 的浓度。黑色的垂直线为 SiO₂/Si 界面。注意：在界面处 B 被耗尽，这归因于氧化物生长过程中 B 的分凝。



B.12 热退火, Drive-in, Activation, and Screening Oxide Strip

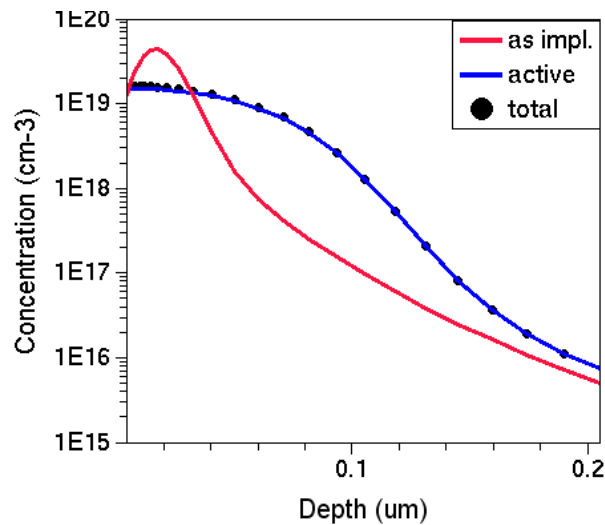
为了消除注入过程引起的缺陷（退火），或者使杂质进一步向衬底方向扩散，或者使注入的杂质活化：

```
diffuse temperature=1000<C> time=30<min>
```


strip oxide

```
SetPlxList { BTotal BActive AsTotal AsActive }
```

```
WritePlx 1Danneal.plx
```



上图比较了注入之后以及退火之后 As 的掺杂分布。从图中可以看出，所有的 As 均已活化。

C. 二维工艺模拟 45min

C.1 简介

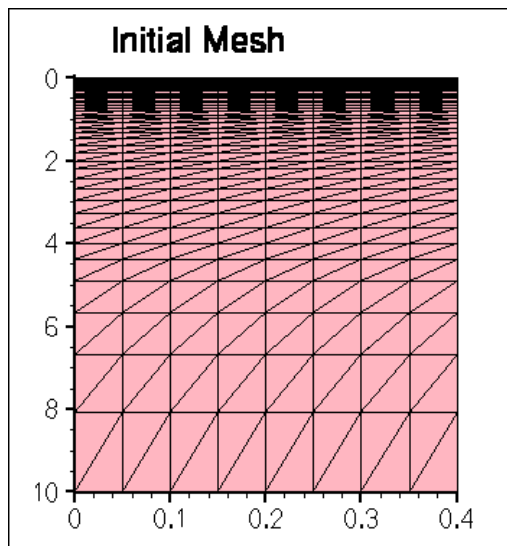
介绍了 0.18um NMOSFET 的二维器件模拟。

C.2 定义初始的二维网格

```
line x location= 0.0      spacing= 1.0<nm>  tag=SiTop
line x location=50.0<nm>  spacing=10.0<nm>
line x location= 0.5<um>  spacing=50.0<nm>
line x location= 2.0<um>  spacing= 0.2<um>
line x location= 4.0<um>  spacing= 0.4<um>
line x location=10.0<um>  spacing= 2.0<um>  tag=SiBottom
```

```
line y location=0.0      spacing=50.0<nm>  tag=Mid
line y location=0.40<um> spacing=50.0<nm>  tag=Right
```

以上这段命令生成如下的二维网格：



注：

- 在一维、二维和三维模拟时，Sprocess 采用的坐标系统是一致的，亦即 x 轴为与 wafer 垂直的方向，y 轴为与 wafer 平行的方向。
- 在遇到第一个掩模之前，Sprocess 采用一维模拟。
- 建议从一个较粗的网格开始模拟，这样可以提高速度。在模拟的过程中，自动网格化将增加网格节点。

C.3 模拟域与初始化

```
region silicon xlo=SiTop xhi=SiBottom ylo=Mid yhi=Right
init concentration=1.0e+15<cm-3> field=Phosphorus wafer.orient=100
```

C.4 B 的注入

```
implant Boron dose=2.0e13<cm-2> energy=200<keV> tilt=0 rotation=0
implant Boron dose=1.0e13<cm-2> energy= 80<keV> tilt=0 rotation=0
implant Boron dose=2.0e12<cm-2> energy= 25<keV> tilt=0 rotation=0
```

第一轮的高能注入用于制作 p 阱，第二轮的中能注入用于制作一个倒向分布的 B 杂质分布以抑制穿通，第三轮的低能注入用于调整阈值电压 V_t 。

C.5 生长栅氧

```
mgoals on min.normal.size=1<nm> max.lateral.size=2.0<um> \
normal.growth.ratio=1.4 accuracy=2e-5
```

```
diffuse temperature=850<C> time=10.0<min> O2
```

```
grid remesh
```

```
select z=Boron
```

```
layers
```

grid remesh 命令强制在新生成的氧化物层中创建 MGOALS 网格。

C.6 制作多晶硅栅

```
deposit poly type=anisotropic thickness=0.18<um>
```

```
mask name=gate_mask left=-1 right=90<nm>
```

```
etch poly type=anisotropic thickness=0.2<um> mask=gate_mask
```

```
etch oxide type=anisotropic thickness=0.1<um>
```

注：

- 关键词 `type=anisotropic` 意味着多晶硅只在纵向上生长。
- 使用掩模的命令包括 `implant`、`etch` 和 `deposit`。

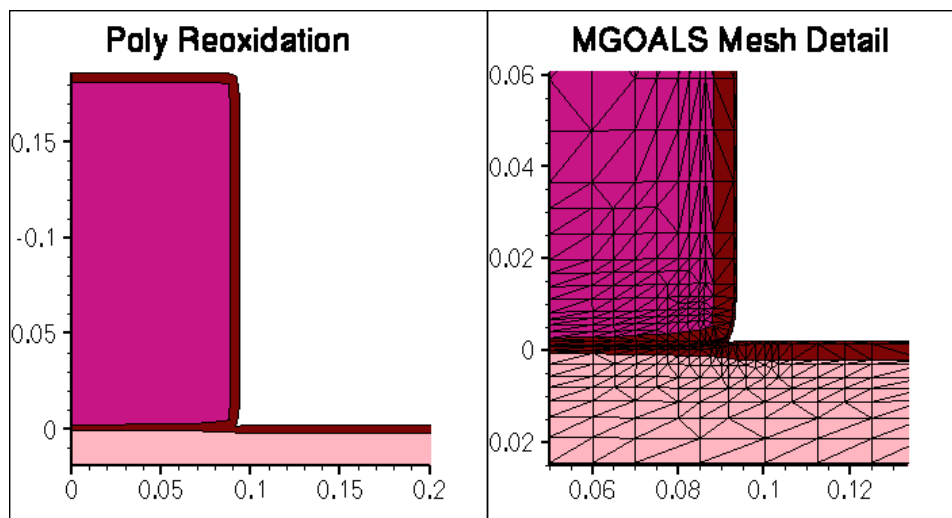
C.7 多晶硅重氧化

为了释放应力，在形成边墙之前生长一层薄的氧化物。

```
diffuse temperature=900<C> time=10.0<min> O2 pressure=0.5<atm> \
    mgoals.native
```

In all diffusion steps, Sentaurus Process accounts for a very thin native oxide layer, which is always present on silicon and quickly forms on newly created interfaces. The keyword `mgoals.native` specifies that MGOALS should be used to deposit this layer. This is **recommended** for complex geometries.

MGOALS 自动在薄的氧化层以及邻近的 Si 与多晶硅中进行网格重定义。



多晶重氧化之后的结构

MGOALS 自动生成的网格

C.8 保存 Snapshots

```
struct tdr=NMOS4
```

以上命令生成 `NMOS4_fps.tdr` 文件，可以采用 `tdx` 查看 `.tdr` 文件，如：

```
%tdx NMOS4_fps.tdr
```

以 df-ise 格式保存 snapshot:

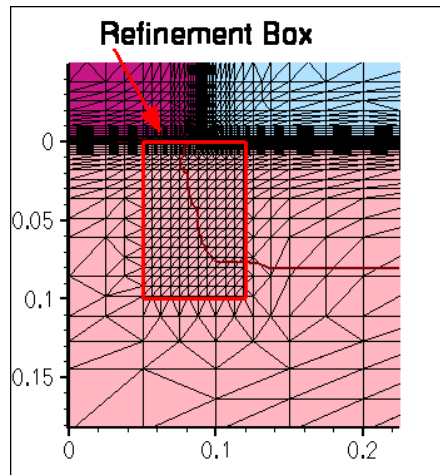
```
struct dfise=NMOS4
```

C.9 在 LDD 和 Halo 注入之前进行网格重定义

```
refinebox silicon min= {0.0 0.05} max= {0.1 0.12} \
    xrefine= {0.01 0.01 0.01} yrefine= {0.01 0.01 0.01} add
refinebox remesh
```

注:

- 在 LDD 区域的掺杂分布是非常重要的, 因此需要使用更加精细的网格。
- xrefine 中的三个值分别指定顶部、中间和底部的网格间距。
- 关键词 add 将网格重定义方框增加到当前的网格策略。



C.10 LDD 和 Halo 注入

```
implant Arsenic dose=4e14<cm-2> energy=10<keV> tilt=0 rotation=0
```

```
implant Boron dose=0.25e13<cm-2> energy=20<keV> \
    tilt=30<degree> rotation=0
```

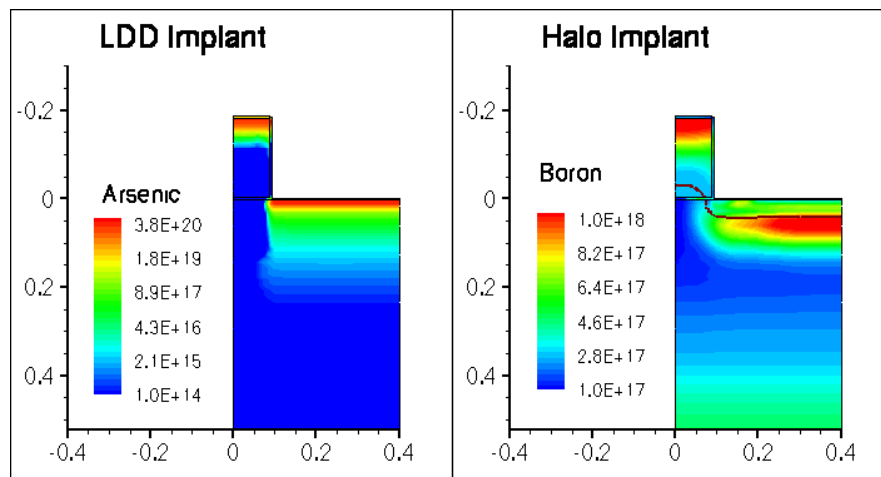
```
implant Boron dose=0.25e13<cm-2> energy=20<keV> \
    tilt=30<degree> rotation=90<degree>
```

```
implant Boron dose=0.25e13<cm-2> energy=20<keV> \
    tilt=30<degree> rotation=180<degree>
```

```
implant Boron dose=0.25e13<cm-2> energy=20<keV> \
    tilt=30<degree> rotation=270<degree>
```

```
diffuse temperature=1050<C> time=5.0<s>
```

- LDD 注入采用较高的剂量 ($4 \times 10^{14} \text{ cm}^{-2}$) 和较低的能量。Halo 注入也采用了较高的剂量 ($1 \times 10^{14} \text{ cm}^{-2}$)。
- 采用快速热退火 (RTA) 使杂质活化。



C.11 形成边墙

```
deposit nitride type=isotropic thickness=60<nm>
etch nitride type=anisotropic thickness=84<nm>
etch oxide type=anisotropic thickness=10<nm>
```

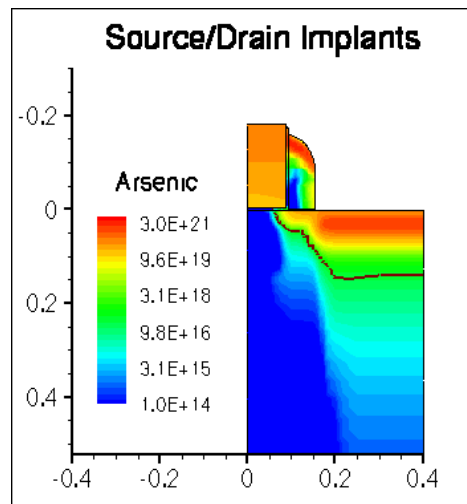
C.12 在源/漏注入之前进行网格重定义

```
refinebox silicon min= {0.04 0.05} max= {0.18 0.4} \
xrefine= {0.01 0.01 0.01} yrefine= {0.05 0.05 0.05} add
refinebox remesh
```

C.13 源/漏注入

```
implant Arsenic dose=5e15<cm-2> energy=40<keV> \
tilt=7<degree> rotation=-90<degree>
diffuse temperature=1050<C> time=10.0<s>
```

- 为了保证源/漏区域具有较低的电阻，采用了非常高的剂量($5 \times 10^{15} \text{ cm}^{-2}$)。采用 7 度倾角以降低沟道效应 (channeling)，-90 度的旋转保证入射平面与栅平行，从而 7 度倾角不会造成源和漏的不对称。
- 沟道效应对单粒子响应是否会有什么影响？比如：产生的缺陷不同，从而影响 SRH 复合，导致电荷收集的不同。（待研究）



C.14 Contact Pads

deposit Aluminum type=isotropic thickness=30<nm>

mask name=contacts_mask left=0.2<um> right=1.0<um>

etch Aluminum type=anisotropic thickness=0.25<um> mask=contacts_mask

etch Aluminum type=isotropic thickness=0.02<um> mask=contacts_mask

C.15 保存整个结构

transform reflect left

struct smesh=NMOS

- 关键词 smesh 创建网格化引擎 Mesh 或者器件结构编辑器 sde 所需的文件。（注：工艺模拟的下一步一般是网格化，或者导入到 sde）
- 工艺模拟的结构也可直接进行器件模拟，这样做是否会有什么影响？因为工艺模拟和器件模拟对网格的需求是不同的。
- 除了*_fps.tdr 文件之外，还保存了 Mesh 所需的输入文件：*_bnd.tdr 和*_msh.cmd。
- 也可保存为 df-ise 格式：struct ise.mdraw=NMOS

C.16 提取一维分布

可以在工艺模拟流程的任何时刻提取一维分布。

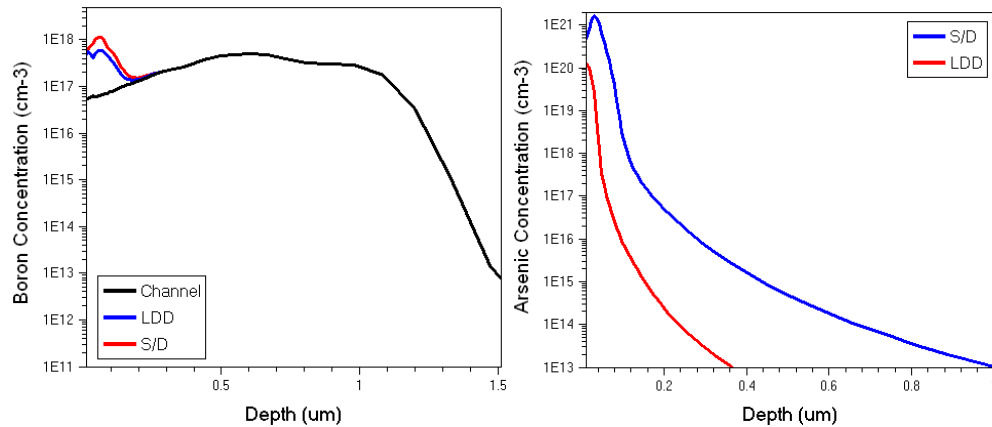
SetPlxList {BTotal NetActive}

WritePlx NMOS_channel.plx y=0.0 silicon

SetPlxList {AsTotal BTotal NetActive}

WritePlx NMOS_Idd.plx y=0.1 silicon

SetPlxList {AsTotal BTotal NetActive}

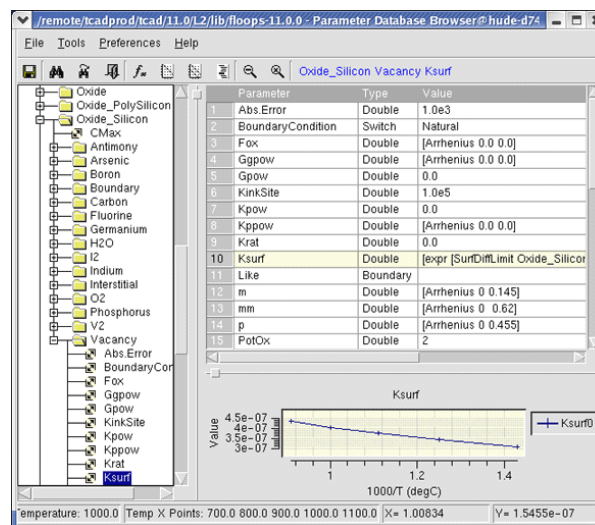
WritePlx NMOS_sd.plx $\gamma=0.35$ silicon

D. 定义模型和指定参数 30min

D.1 属性数据库浏览器

/remote/tcadprod/tcad/11.0/L1/lib/floops-11.0.0/Params 用于存储默认模型和参数 (Sentaurus 安装完之后自带)

```
%sprocess --pdb
```



D.2 在输入文件中改变参数

```
pdbSet Oxide_Silicon Vacancy Ksurf {[Arrhenius 1e3 0.1]}
```

E. 采用全定制的校准文件 30min

E.1 简介

校准文件用于保存所有的模型选择和参数设置。

在 Sprocess 输入文件的开头导入校准文件：source calibration.fps

E.2 全局设置

```
set ProcessStepID "Global Settings"
```

```
SetDiosPairModelMode
```

```
pdbSetSwitch Silicon Boron ActiveModel Transient
```

```
pdbSetDouble Silicon Boron Solubility \
                                     {[Arr 2e22 0.4593]}
```

```
pdbSetDoubleArray Silicon Boron Interstitial D \
```

```
{ -1 0.0
  0 {0.92 * [Arr 3.3e-2 3.260]}
  1 {0.92 * [Arr 1.25 3.625]}
  2 0.0 }
```

- SetDiosPairModelMode 选择 charged pair（可以理解为电子空穴对）的扩散模型与 Dios 中的电子空穴对扩散模型兼容。

F. 采用高级校准 30min

F.1 简介

高级校准为用户提供一套面向深亚微米工艺校准过的参数，包括超浅结的形成、表面剂量损失、沟道和 halo 注入与扩散。

F.2 激活高级校准

在输入文件的开头，插入：

```
AdvancedCalibration
```

注：

- 在工艺模拟时选择模型和参数具有相当的难度。
- 采用高级校准之后对器件的单粒子响应是否存在影响？

G. 工艺模拟与 SWB 和 Ligament 30min

G.1 简介

Ligament 流程编辑器提供一个 GUI 界面用于创建和编辑工艺流程。用户可以在 ligament 环境下，采用宏（包括用户定义的局部宏和 Ligament 工具库提供的全局宏）组装一个工艺流程。Ligament 采用 SPR (Simple Process Representation) 语言，该语言独立于任何工艺模拟器，从而，利用 Ligament 组装的工艺流程可以采用任何工艺模拟器进行模拟。

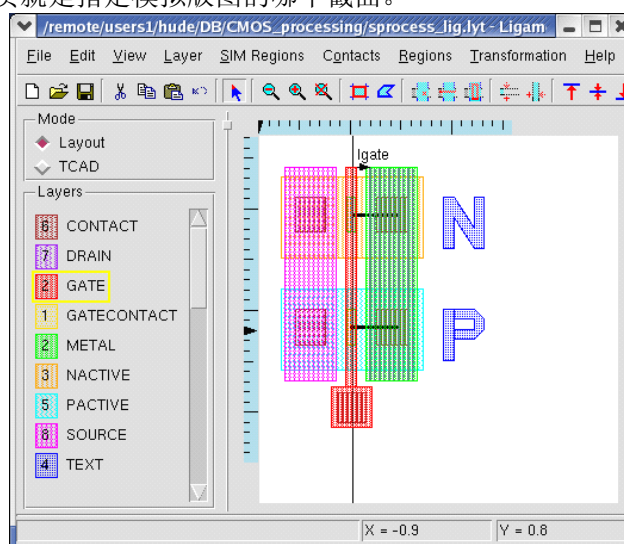
但是，某些与模拟器相关的控制序列并不是由 SPR 直接处理的。本节将介绍如何在 Ligament SPR 流程中包含与 Sprocess 或者 swb 相关的指令。

G.2 参数化版图

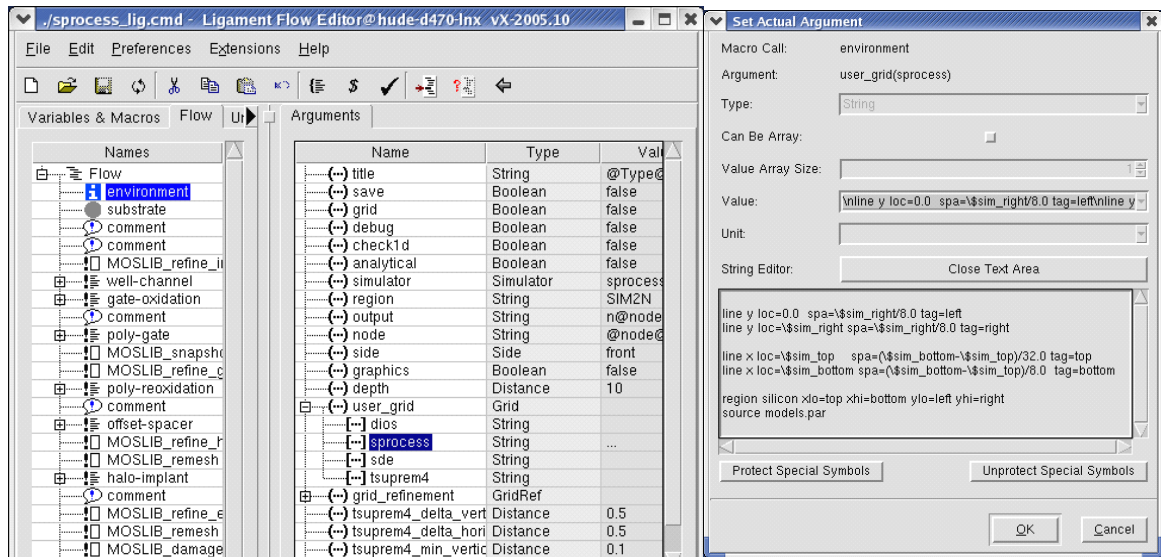
SentaurusP											
		Type	CUT	lgate	f	ver	Poly	Halo	Ext	SD	Backend
1	--	nMOS	SIM2N	0.09	2	1	0	0	0	0	0
2				0.05	2	1	0	0	0	0	0
3		pMOS	SIM2P	0.09	2	1	0	0	0	0	0
4				0.05	2	1	0	0	0	0	0

Sprocess 使用的参数包括：

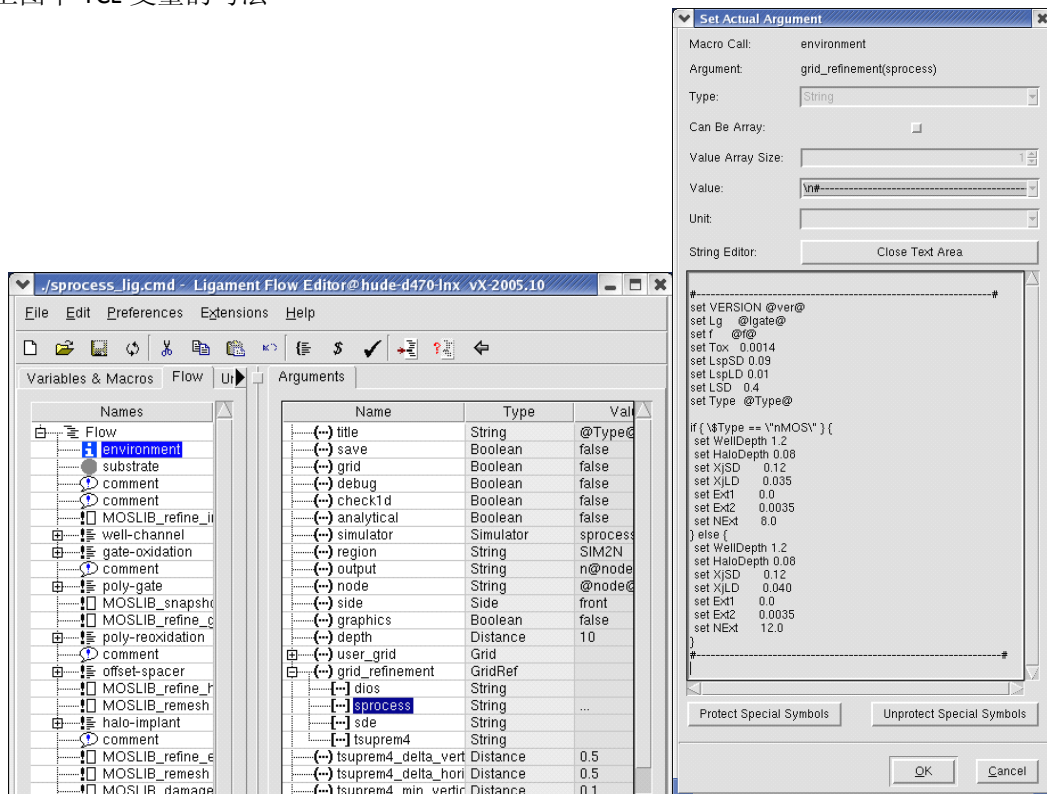
- Type = nMOS|pMOS 用于指定是创建 NMOS 还是 PMOS 器件。
- CUT = SIM2N|SIM2P 指定将二维掩模转换为二维掩模时使用的割线（版图是二维的，工艺模拟时的掩模是一维的）。其实就是指模拟版图的哪个截面。



G.3 定义网络



注意右上图中 TCL 变量的写法



注意右上图中 Sprocess 命令文件引用 SWB 参数的方法，即采用@。

H. Custom Models with Alagator 45min

H.1 简介

在 Sprocess 中，可以采用 Alagator（A Layered Approach to Generating Advanced Technically-Oriented

I. Special Focus: 槽刻蚀 35min

I.1 简介

本节讨论 STI 的工艺流程。

1. Growing pad oxide
2. Depositing nitride layer
3. STI lithography
4. Shallow trench etch
5. Growing oxide liner
6. Depositing and CMP of TEOS
7. Nitride strip/reflect

I.2 Initialization

```

line y location=0.0      spacing= 0.1<um> tag=left
line y location=0.5<um> spacing= 0.1<um> tag=right

line x location=0.0      spacing= 5.0<nm> tag=bottom
line x location=0.5<um> spacing=50.0<nm>
line x location=5.0<um> spacing= 0.5<um> tag=top

region Silicon xlo=bottom xhi=top ylo=left yhi=right
init  concentration=1.4e+15<cm-3> field=boron wafer.orient=100

```

I.3 生长 Pad Oxide

```

gas_flow clear
gas_flow name=O2_0.1_N2_10 pressure=1.0<atm> \
    flowO2=0.1<l/min> flowN2=10.0<l/min>
gas_flow name=O2 pressure=1.0<atm> \
    flowO2=1.0<l/min>

temp_ramp name=PadOxide clear
temp_ramp name=PadOxide time=(1050.0-700.0)/75<s> \
    temp= 700.0<C> ramprate=75<K/s> gas_flow=O2_0.1_N2_10
temp_ramp name=PadOxide time=1.5<min> \
    temp=1050.0<C> hold gas_flow=O2
temp_ramp name=PadOxide time=(1050.0-700.0)/20<s> \
    temp=1050.0<C> ramprate=-20<K/s> gas_flow=O2_0.1_N2_10

```

```
diffuse temp_ramp=PadOxide
```

```
set PadOxThick [MeasureOx Silicon 2 0.0]
puts "Thickness if PadOx is: $PadOxThick um"
```

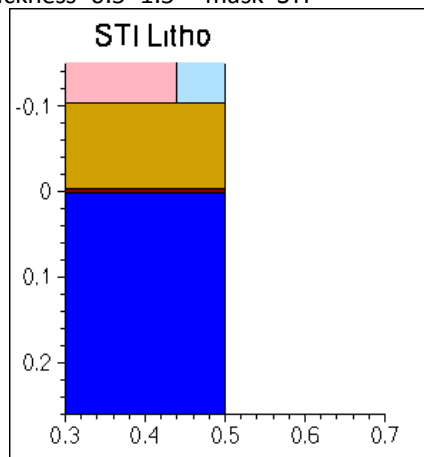
- MeasureOx 三个参数的含义分别为：（1）氧化层的下层材料；（2）模拟的维数；（3）测量氧化物厚度的位置（y 坐标）。

I.4 淀积氮化层

```
machine name=NiDep nitride deposit isotropic rate=1.0<um/min>
set NitrideThick 0.1
deposit machine=NiDep time=$NitrideThick<min>
```

I.5 STI 光刻

```
mgoals on
set TrenchLeftCoord 0.44
set TrenchRightCoord 0.56
mask name=STI segments= {-1.0 $TrenchLeftCoord $TrenchRightCoord 2.0}
deposit Photoresist isotropic thickness=0.5
etch Photoresist anisotropic thickness=0.5*1.5 mask=STI
```



I.6 浅槽刻蚀

```
proc etchAngle { Angle Material Depth } {

set alpha [expr ${Angle}*atan(1.0)/45.0] ; #Degree to radiant conv.
set x1 [expr sin($alpha)] ; #x-component of etch directional vector
set x2 [expr cos($alpha)] ; #y-component of etch directional vector
set x3 0 ; #z-component of etch directional vector
set etchRate [expr 1.0/sin($alpha)]
```

```
etch material=$Material time=$Depth type=directional \
    direction= { $x1 $x2 $x3 } rate=$etchRate
}
```

上面这段命令定义了一个过程，该过程包括三个参数：刻蚀的角度、材料和深度。

```
set NitrideAngle 87.0
```

```
etchAngle $NitrideAngle Nitride $NitrideThick*1.5
```

```
etch Oxide anisotropic thickness=$PadOxThick*1.5
```

```
strip Photoresist
```

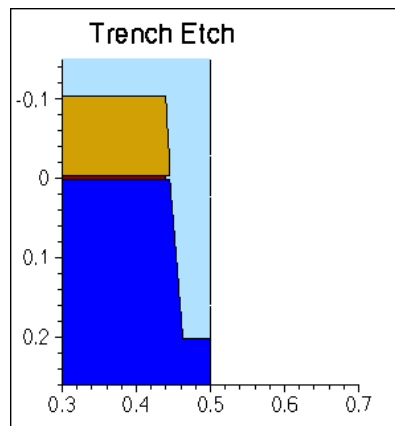
```
set TrenchAngle 85.0
```

```
set TrenchDepth 0.2
```

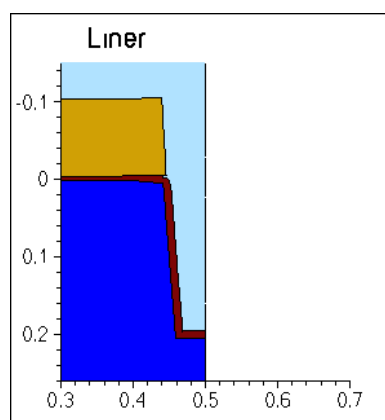
```
etchAngle $TrenchAngle Silicon $TrenchDepth
```

```
etch Oxide isotropic thickness=0.005
```

上面这段命令以 87 度倾角刻蚀氮化物→氧化物的各向同性刻蚀→以 85 度倾角刻蚀 Si→氧化物的各向同性刻蚀。



1.7 生长衬垫氧化物

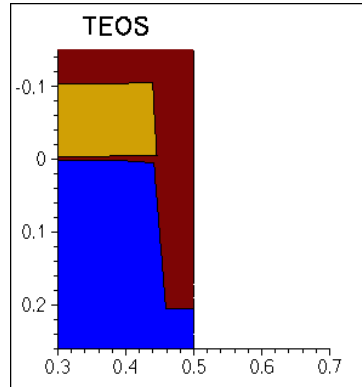


I.8 TEOS 的淀积与 CMP

deposit Oxide isotropic thickness=0.10

etch material= {Oxide} type = cmp coord=-0.05

etch material= {nitride} type = cmp coord=-0.05



I.9 氮化物的剥离/反射

strip Nitride

transform reflect right

struct tdr=STI

J. Special Focus: Meshing with MGOALS(15 分钟)

J.1 简介

MGOALS 执行两个动作：边界移动和网格优化。本节关注网格优化。

采用 min.normal.size、normal.growth.ratio 以及 max.lateral.size 设置垂直和水平方向的网格尺寸。

J.2 Initialization

line x location= -0.15 spacing=0.01 tag=GateTop

line x location= -0.02 spacing=0.005

line x location= -0.01 spacing=0.005 tag=OxTop

line x location= 0.0 spacing=0.005 tag=SiTop

line x location= 0.05 spacing=0.01

line x location= 0.5 spacing=0.05

line x location= 1.0 spacing=0.2 tag=SiBottom

line y location= 0.0 spacing=0.05 tag=Mid

line y location= 0.1 spacing=0.05 tag=Gate


```
line y location= 0.4 spacing=0.05 tag=Right
```

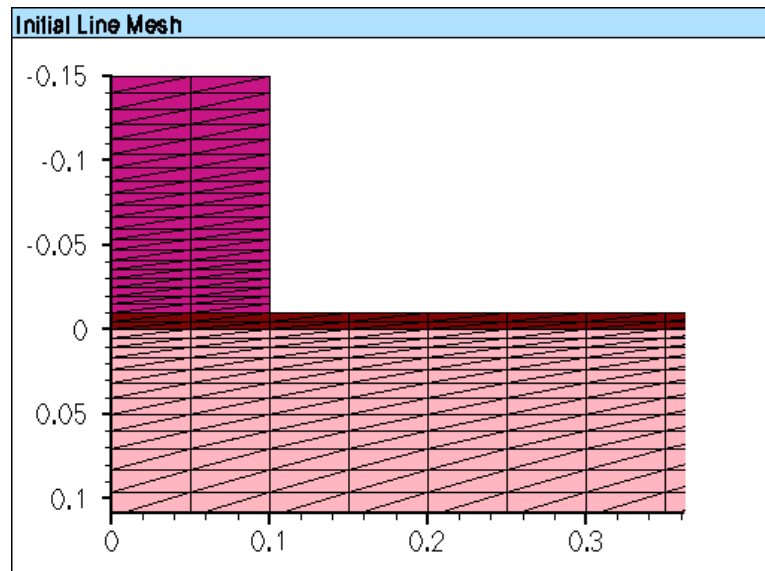
```
region silicon xlo=SiTop xhi=SiBottom ylo=Mid yhi=Right
```

```
region oxide xlo=OxTop xhi=SiTop ylo=Mid yhi=Right
```

```
region poly xlo=GateTop xhi=OxTop ylo=Mid yhi=Gate
```

```
struct tdr=grid_mg0
```

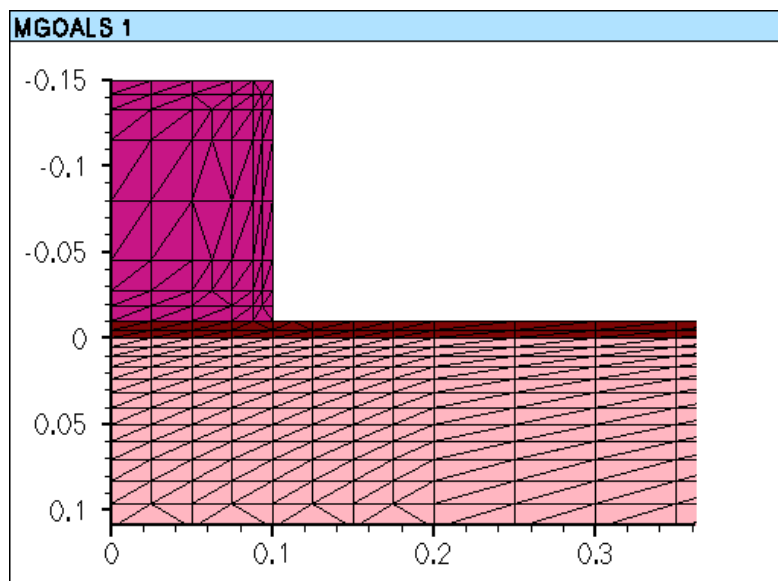
以上命令生成的初始网格见下图：



J.3 MGOALS Remesh

```
mgoals on min.normal.size=0.01 normal.growth.ratio=2.0
```

```
grid remesh
```



MGOALS 在所有的界面产生 boundary-conforming 网格。min.normal.size 设置垂直方向的网格间距。沿着远离界面的方向，网格间距不断变大，变化速率由 normal.growth.ratio 控制。

J.4 MGOALS 网格优化框

```
refinebox min= {-0.1 0.05 } max= { 0.1 0.15 } \
          xrefine= {0.015} yrefine= {0.015} all add
grid      remesh
```

其中 all 指定对框中所有材料进行网格优化。

K. Special Focus: 3D 工艺模拟（Sprocess 与 sde 结合）(30 min)

目标：阐述在 Sprocess 中如何采用 SDE 库进行三维边界移动操作。

K.1 简介

SDE 已经被集成到 Sprocess 中，以便使用 sde 的三维边界移动算法。

Sprocess 在三个方向上以相似的方法执行淀积、刻蚀以及相关的操作。MGOALS 库在一维和二维模拟执行这些操作。SDE 执行三维操作，包括掩模、刻蚀、淀积、photo、剥离和 transform。Sprocess 在三维与二维和一维之间自动切换。该模式的详细阐述以及示例可参考“[Sentaurus Process User Guide](#)”第 11 章。

Sprocess 产生的结构包括空气层（gas）。

K.2 各项异性刻蚀

```
pdbSet Mechanics EtchDepoRelax 0
```

```
line x loc=-0.25 spac=0.05
line x loc=0.0    spac=0.01 tag=substop
line x loc=0.05   spac=0.05
line x loc=0.5    spac=0.1  tag=subsbottom
```

```
line y loc=0.0 tag=ylo spac=0.1
line y loc=0.6 tag=yhi spac=0.1
```

```
line z loc=0 tag=zlo spac=0.1
line z loc=0.4 tag=zhi spac=0.1
```

```
region silicon xlo=substop xhi=subsbottom
```

```
mgoals on min.normal.size=0.02 accuracy=1e-6 max.box.angle=165 normal.growth.ratio=2
max.lateral.size=0.3
refinebox interface.materials = {Silicon}
refinebox min= {-10 -10 -10 } max= {10 10 10} xrefine=0.3 yrefine=0.3 zrefine=0.3 add
```

```
init
deposit oxide type=fill coord=-0.002

# Note! deposition with a mask deposits where there is no mask
mask name=nimask left=0.3 right=1 front=0.2 back=1
sde logfile=AnisoEtch.scm on
pdbSet InfoDefault 1
deposit nitride mask=nimask thickness=0.15
struct tdr=etch0
```

```
etch oxide thickness=0.5<um> anisotropic info=1
struct tdr=etch1
```

```
etch silicon thickness=0.3<um> anisotropic info=1
struct tdr=AnisoEtch
```

在遇到第一个三维掩模之后，Sprocess 自动转换到 sde 模式。

K.3 各向同性淀积

```
pdbSet Mechanics EtchDepoRelax 0
```

```
line x loc=-0.25 tag=gastop spac=0.05
line x loc=0.0 tag=substop spac=0.01
line x loc=1.5 tag=subsbottom spac=0.2
```

```
line y loc=0.0 tag=ylo spac=0.1
line y loc=0.3
line y loc=0.6 tag=yhi spac=0.1
```

```
line z loc=0 tag=zlo spac=0.1
line z loc=0.2 spac=0.01
line z loc=0.4 tag=zhi spac=0.1
```

```
region silicon xlo=substop xhi=subsbottom ylo=ylo yhi=yhi zlo=zlo zhi=zhi
region gas xlo=gastop xhi=substop ylo=ylo yhi=yhi zlo=zlo zhi=zhi
```

```
mgoals min.normal.size=0.02 accuracy=1e-4 max.box.angle=165 normal.growth.ratio=4
refinebox min= {-10 -10 -10} max= {10 10 10} xrefine=0.2 yrefine=0.2 zrefine=0.2
refinebox interface.materials = {Silicon Oxide}
init
sde logfile=depo.scm on
pdbSet InfoDefault 1
```

```
#
# NOTE!
# Mask specifications, as well as specifications of cuboids etc
# inside of sde commands are supposed to be in sde (aka dfise) coordinates
# floops mask commands are in floops coordinates.
# The difference is: floopsX --> -sdeZ and floopsZ --> sdeX
#
sde {
  (sdegeo:set-default-boolean "ABA" )
  (define r1 (sdegeo:create-cuboid (position 0 0.6 0)
    (position 0.2 0.3 0.5) "Silicon" "Silicon_2" ))
  ( define facelist (list (car (find-face-id (position 0.1 0.3 0.25)))
    (car (find-face-id (position 0.2 0.5 0.25)))) )
  (sdegeo:taper-faces facelist (position 0.2 0.3 0.5) (gvector 0 0 1) 3)
  (sdeio:save-dfise-bnd "all" "out1_sde.bnd")
}

deposit oxide thickness=0.07 isotropic info=1
struct tdr=Depo_final
注意：Sprocess 命令文件中嵌入了 sde 脚本。
```

K.4 Multimaterial Etching

```
pdbSet Mechanics EtchDepoRelax 0
```

```
line x loc=-0.4 tag=gastop      spac=0.05
line x loc=-0.3 tag=potop       spac=0.05
line x loc=0. tag=sitop         spac=0.01
line x loc=0.3                  spac=0.03
line x loc=1.0 tag=subsbottom
```

```
line y loc=0.0 tag=ylo         spac=0.1
line y loc=0.5 tag=ymid        spac=0.1
line y loc=1.0 tag=yhigh       spac=0.1
```

```
line z loc=0.0 tag=zlo         spac=0.1
line z loc=0.5 tag=zmid        spac=0.1
line z loc=1.0 tag=zhigh       spac=0.1
```

```
region silicon xlo=sitop xhi=subsbottom ylo=ylo yhi=yhigh zlo=zlo zhi=zhigh
region oxide xlo=potop xhi=sitop ylo=ymid yhi=yhigh zlo=zmid zhi=zhigh
region nitride xlo=potop xhi=sitop ylo=ylo yhi=ymid zlo=zmid zhi=zhigh
region polysilicon xlo=potop xhi=sitop ylo=ymid yhi=yhigh zlo=zlo zhi=zmid
region silicon xlo=potop xhi=sitop ylo=ylo yhi=ymid zlo=zlo zhi=zmid
```

```
region gas      xlo=gastop xhi=potop ylo=ylo yhi=yhigh zlo=zlow zhi=zhigh
```

```
init
```

```
struct tdr=init
```

```
mgoals on min.normal.size=0.02 accuracy=1e-6 max.box.angle=165 \
```

```
normal.growth.ratio=2 max.lateral.size=0.3
```

```
refinebox interface.materials = {Silicon}
```

```
refinebox min= {-10 -10 -10} max= {10 10 10} xrefine=0.3 yrefine=0.3 \
```

```
zrefine=0.3 add
```

```
sde logfile=multimatetch.scm on
```

```
pdbSet InfoDefault 1
```

```
mask name=em left=0.3 right=0.6 front=0.4 back=0.8
```

```
etch time=1 info=1 type=anisotropic material= { oxide silicon nitride} \
```

```
rate= {0.1 0.2 0.28} mask=em sde= {"algorithm" "lopx"}
```

```
struct tdr=final
```

同时刻蚀多种材料。sde= {"algorithm" "lopx"} 指定 sde 采用 lpx 算法。

K.5 Reflect and Clip

```
pdbSet Mechanics EtchDepoRelax 0
```

```
line x loc=-0.25 tag=gastop spac=0.05
```

```
line x loc=0.0 tag=substop spac=0.01
```

```
line x loc=1.5 tag=subsbottom spac=0.2
```

```
line y loc=1.65 spac=0.1
```

```
line y loc=1.95 spac=0.1
```

```
line z loc=0.15 spac=0.1
```

```
line z loc=0.6 spac=0.1
```

```
region silicon xlo=substop xhi=subsbottom
```

```
mgoals min.normal.size=0.02 accuracy=1e-4 max.box.angle=165 normal.growth.ratio=4
```

```
refinebox min= {-10 -10 -10} max= {10 10 10} xrefine=0.2 yrefine=0.2 zrefine=0.2
```

```
refinebox interface.materials = {Silicon Oxide}
```

```
init
```

```
deposit oxide thickness=100 isotropic
```

```
sde logfile=reflect_clip.scm on
```

```
pdbSet InfoDefault 1
```

mask layoutfile=simple.lyt

A layout file "simple.lyt" generated with Prolyt is loaded.

For more information about Ligament and Prolyt, refer to Section 2 of the HTML

training material.

deposition with masks deposits where there is no mask

mask name=Mask2 negative

deposit nitride thickness=0.25 anisotropic info=1 mask=Mask2

etch oxide thickness=120 type=anisotropic mask=Mask1

struct tdr=initial

transform reflect left

struct tdr=reflect

transform clip min= {-10 1.35 0.15} max= {10 1.65 0.4}

struct tdr=clip

注意版图文件的导入以及 mask 的使用。版图文件中定义了 Mask1 和 Mask2。

SDE（结构编辑器）

A. Getting Started（15 分钟）

目标：

- 熟悉 GUI 的特性
- 熟悉 SDE 使用的文件类型
- 产生一个简单的器件结构

A.1 简介

SDE 是一个二维和三维器件编辑器以及三维工艺仿真器。具有三种截然不同的工作模式：二维结构编辑、三维结构编辑和三维工艺仿真。

利用 GUI，采用二维或三维元件（例如：矩形、多边形、立方体、圆柱体、球体）以几何形式创建二维和三维器件。

可以通过 filleting，3D edge blending 和 chamfering 产生圆化的边缘。可以采用基本元件通过简单的相交产生复杂的形状。

GUI 包含一个命令行窗口，窗口中即时显示 GUI 操作的菜单命令。

在工艺仿真模式下，SDE 将工艺步骤（刻蚀与淀积、patterning、填充与抛光）转化为几何操作。Procem 支持各种选项，例如各向异性和各向同性刻蚀与淀积、rounding 和 blending。

A.2 启动 SDE

B2500-4% sde &

A.4 输入/输出文件类型

SDE 支持的输入/输出文件的主要类型包括：

- 脚本文件（.scm）

用户定义的脚本文件，可用于以批处理的方式建立一个器件结构。

- 日志文件（.jrl）

该文件除了包含脚本命令之外，还包含日志命令。日志命令控制脚本文件的执行。例如，以下命令：

(journal:step #t)

指示 SDE 停止脚本的连续执行，脚本的剩余部分一行一行地单步执行。

- ACIS SAT 文件（.sat）

该文件包含模型的几何结构（ACIS 格式），不能直接编辑。

- DF-ISE 边界文件（.bnd）

这是一个 DF-ISE 格式的边界文件。该文件可以直接被加载到 SDE 或者其它 Synopsys TCAD 工具，如网格

化引擎 Mesh 和 Noffset2D/3D。

注：边界文件也可以 TDR 格式（.tdr）创建。

● DF-ISE 掺杂和优化文件（.cmd）

这是一个 DF-ISE 格式的文件，包括掺杂和网格优化信息，结合对应的边界文件，可唯一地定义模型的几何结构。

注 1：当执行命令 **File->Save Model** 或 **File->Save Model As** 时，SDE：（1）以 ACIS 格式保存模型的几何结构（.sat）（2）保存 Ref/Eval 窗口和参数（.scm）（3）网格优化和掺杂相关的信息（.cmd）（4）DF-ISE 边界文件（.bnd）。

注 2：TDR 边界文件可以采用以下命令查看：

```
(sdeio:save-tdr-bnd (get-body-list) "name_of_the_file.tdr")
```

在一个终端窗口中键入以下命令，可以在 **TDR 文件查看器** 中打开 TDR 文件：

```
B2500-4%tdx "name_of_the_file.tdr"<
```

A.5 创建一个简单的结构

SDE 提供两种不同的方法用于创建和编辑器件结构：

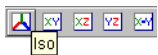
- 基于 GUI 的方法
- 基于脚本的方法

除此之外，SDE 还具备另一种非常灵活的方法。该方法接受 GUI 和脚本的混合输入，且可以使用命令行窗口键入命令。由于 GUI 操作时，命令行窗口有对应的脚本显示，所有以前执行过的命令，无论是基于 GUI 的还是基于脚本的，均可被再次调用、编辑和重新执行。

示例

创建一个包含一个 Si 立方体和一个氧化物立方体的结构：

- 1、采用 **File->New** 重新初始化 SDE（推荐）
- 2、在 Material 列表框中选择 Silicon
- 3、单击 ISO 图标转换到 3D 立方视图



- 4、**Draw->Exact Coordinates** 转换到 Exact Coordinates 模式
- 5、**Draw->Create 3D Regions->Cuboid**
- 6、在显示窗口中拖动鼠标，释放鼠标后，显示以下对话框：

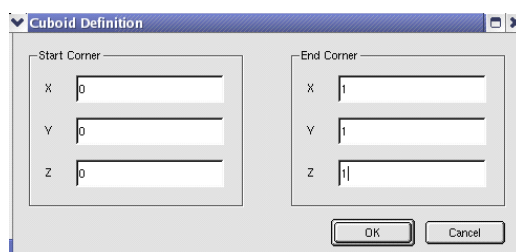


图 2 立方体定义对话框

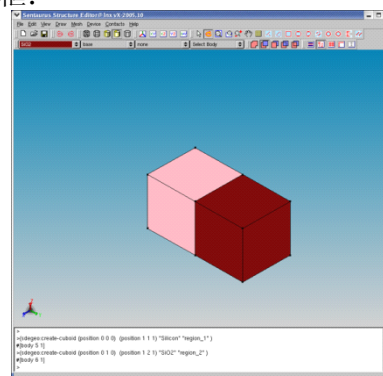


图 3 GUI 中显示的立方体

- 7、在起始点和终止点分别输入(0 0 0)和(1 1 1)

命令行窗口中出现的最后脚本为：

```
(sdegeo:create-cuboid (position 0 0 0) (position 1 1 1) "Silicon" "region_1")
```

在命令行窗口中输入：

```
(sdegeo:create-cuboid (position 0 1 0) (position 1 2 1) "SiO2" "region_2")
```

最终结果如图 3 所示

B. 生成二维边界（45 分钟）

B.1 简介

本节介绍如何一步一步地生成一个如图 1 所示的二维 SOI MOSFET 器件。

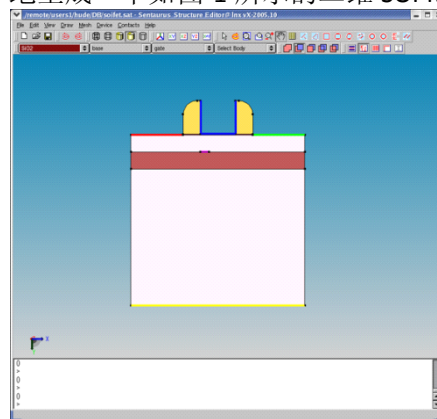


图 1 二维 SOI MOSFET 器件

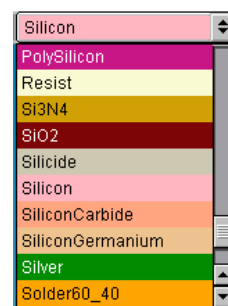


图 2 Material 列表框

B.2 重新初始化 SDE

File->New

B.3 Exact Coordinates 模式

Draw > Exact Coordinates

B.4 选择材料

使用 Material 列表框，见图 2。

B.5 选择默认的布尔表达式

在大多数情况下，一个器件包括多个区域。在采用 SDE GUI 构建器件时，后添加的区域可能与已经存在的区域重叠。重叠的部分到底是哪种材料？这需要一个预先定义的方案。

SDE 包括多种预先定义的方案。例如：采用 New Replaces Old 方案时，新添加的区域将替换已经存在的区域。

Draw->Overlap Behavior->New Replaces Old，或者单击工具栏的相应按钮。



B.6 创建矩形区域

按图 3 的参数创建衬底。

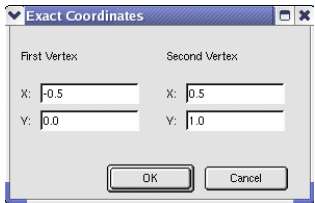


图 3 Exact Coordinates 对话框

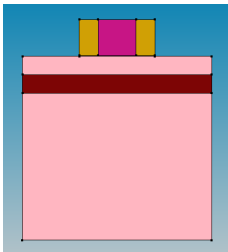


图 4 器件结构

按下表所示数据分别创建栅氧、氮化物边墙、多晶硅和埋氧。

Region	Material	Coordinates
Gate oxide	SiO2	(-0.2 -40e-4), (0.2 0.0)
Nitride spacer	Si3N4	(-0.2 -0.2), (0.2 -40e-4)
Poly gate	PolySi	(-0.1 -0.2), (0.1 -40e-4)
Buried oxide	SiO2	(-0.5 0.1), (0.5 0.2)

最终结果如图 4 所示。

B.7 创建 Single-Lumped 区域

在本例中，Si 衬底和氮化物边墙首先是一大块矩形区域。然后，这些区域被分成两个互不相连的小块（lump）（中间为埋氧和栅多晶）。

在保存边界文件时，每一个小块（lump）被重定义为一个单独的区域。但是，在没有保存之前，掺杂和网格优化不能应用于区域名，因为最终的区域名在保存之前不可知。为了解决这一问题，可以显示地调用小块重定义程序，手工地对每一个小块进行重命名。

创建 Single-lumped 区域：

Edit->Separate Lumps

新指定的名称为：oldregionname_lump_1, oldregionname_lump_2, 等等，其中，oldregionname 为旧的大块区域的名称。

B.8 圆化边缘

SDE 允许对角和边缘进行圆化。

圆化氮化物边墙的两个外角：

1、Edit->Parameters，弹出参数定义对话框

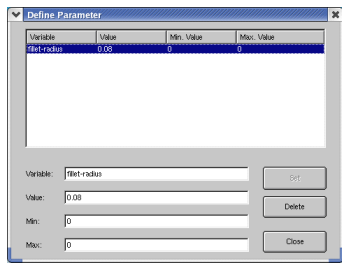


图 5 参数定义对话框

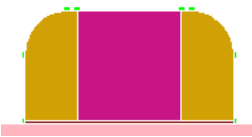
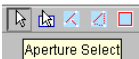


图 6 采用参数 fillet-2d 对边墙氮化物的角进行圆化

- 2、在 Variable 域输入 fillet-radius，在 Value 域输入 0.08
- 3、单击 Set 和 Close
- 4、在 Selection Level 列表框中选择 Select Vertex
- 5、在工具栏中点击 Aperture Select 按钮



- 6、点击边墙氮化物的左上角使其高亮
 - 7、Edit->Edit 2D->Fillet
被选择的角现在被圆化。
 - 8、对右上角进行重复操作
- 结果见图 6：
对应的脚本命令为：

```
(sde:define-parameter "fillet-radius" 0.08 0.0 0.0 )  
(sdegeo:fillet-2d (find-vertex-id (position -0.2 -0.2 0.0)) fillet-radius)  
(sdegeo:fillet-2d (find-vertex-id (position 0.2 -0.2 0.0)) fillet-radius)  
(译者注：熟练之后，使用脚本更加方便快捷。)
```

注意：在命令行窗口中，显示的圆化命令为(sdegeo:fillet-2d-GUI)。在以上给出的脚本中，这一命令被(sdegeo:fillet-2d ...)替换。这一替换允许采用命令(find-vertex-id (position x y z)) 替代鼠标对角进行选择。

B.9 定义接触

通过定义接触可以使器件连接到外部电源。
定义接触：

- 1、Contacts->Contact Sets，显示 Contact Sets 对话框。

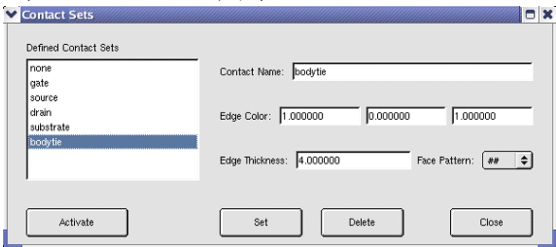


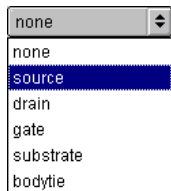
图 7 接触设置对话框

- 2、定义 Contact 的属性。在 Contact Name 域输入接触的名称。
- 3、在 Edge Color 域指定 RGB 数值，修改标记 contact 的 Edge Thickness 值。Face Pattern 域只有在 3D Contact 时才有效。
- 4、点击 set。

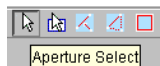
B.10 在已存在的边缘处设置接触

接触只有在与器件结构的一部分关联之后才变得有效。将接触与器件的一个特定的边相关联：

- 1、Contacts->Contact Sets，显示 Contact Sets 对话框
- 2、在 Defined Contact Sets 域中选择已经定义好的接触，如：source 接触
- 3、点击 Activate 激活被选择的接触
- 4、注：接触也可通过 Contact 列表激活



- 5、在 Selection Level 列表中选择 Select Edge。
- 6、点击 Aperture Select 按钮



- 7、点击器件结构中需要定义 source 接触的边。被选择的边高亮显示。

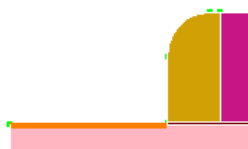


图 8

- 8、Contacts->Set Edge(s)

译者注：总结为三步：

- 1、激活定义好的接触
- 2、选择一条边
- 3、Contacts->Set Edge(s)

对应的脚本命令为：

```
(sdegeo:define-2d-contact (find-edge-id (position -0.4 0.0 0.0)) "source")
(sdegeo:define-2d-contact (find-edge-id (position 0.4 0.0 0.0)) "drain")
(sdegeo:define-2d-contact (find-edge-id (position 0.0 1.0 0.0)) "substrate")
```

注意：命令行窗口中显示的脚本命令(sdegeo:set-current-contact-set "source")和(sdegeo:define-2d-contact-GUI) 已经被(sdegeo:define-2d-contact ...) 替换。这一脚本允许采用(find-edge-id (position x y z))替换鼠标指定被选边的坐标。

B.11 添加顶点

为了模拟二维的 SOI MOSFET，有时需要添加伪体引出。伪体引出通常添加在 Si 外延层和埋氧的界面处，并且是栅下靠近源端一侧。在定义这样的伪体引出之前，需要添加两个额外的顶点。

添加顶点的步骤为：

- 1、Draw->Add Vertex
- 2、点击显示窗口中需要添加顶点的位置，弹出 Exact Coordinates 对话框

3、输入（-0.1 0.1）

4、重复定义第二个顶点（-0.05 0.1）

5、在二维的默认透视模式（Gouraud shaded）下，新添加的顶点很难看见。采用工具栏可以改变透视模式。



6、将 body tie 接触与新定义的边相关联。

图 9 所示为定义了所有接触的器件，透视模式为 Wireframe 模式。

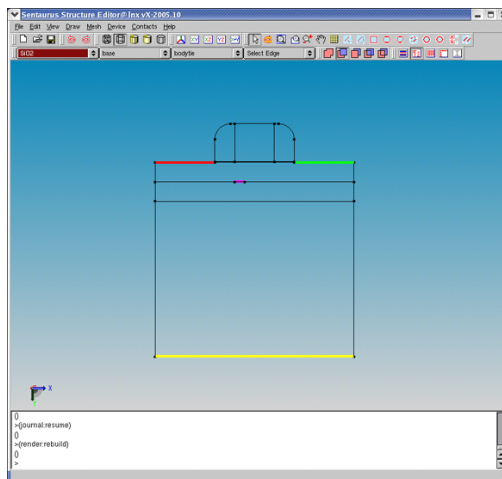


图 9 定义了所有接触的器件（透视模式为 Wireframe 模式）

B.12 定义一个区域为接触

在某些应用时，接触能够覆盖器件的整个二维区域（注：二维器件结构）。例如，MOS 器件的金属和多晶硅区通常将其整体作为一个理想的接触。

定义一个区域为接触的步骤为：

- 1、激活定义好的接触
- 2、在 Selection Level 列表中选择 Select Body
- 3、点击工具栏中的 Aperture Select 按钮
- 4、选择二维区域使其高亮

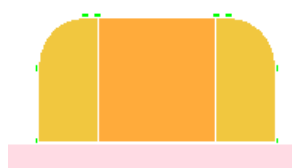


图 10

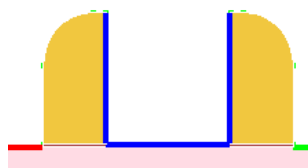


图 11

5、Contacts->Set Region Boundary Edges

这一操作将被选择的区域转换为一个接触。被选区域的边界变为接触定义的颜色和样式。

6、去除栅区域：选择栅区域，Edit->Remove->Region

B.13 对区域进行重命名

鼓励用户进行重命名以方便记忆和引用。

采用 R.Substrate, R.Box, R.Siliconepi, R.Gateox, R.Spacerleft, R.Spacerright 对本例中器件的各个区域进行重命名。

B.14 保存模型

File > Save Model

C. 生成掺杂分布（20 分钟）

C.1 定义材料的掺杂浓度为常数

首先，介绍定义材料背景掺杂浓度的最基本的方法。

在 Si 材料中设置 B 的背景掺杂浓度为 $1 \times 10^{15} \text{cm}^{-3}$ 。

1、Device->Constant Profile Placement, 弹出如下对话框。

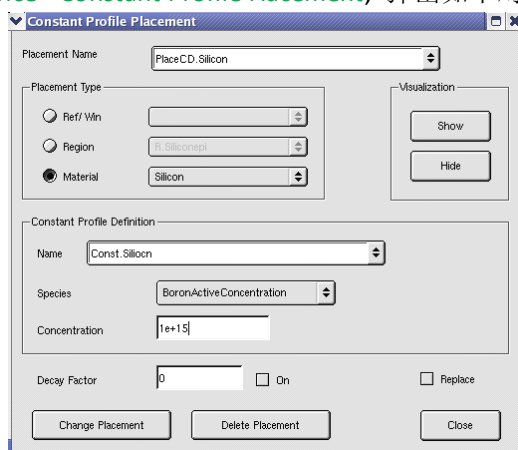


图 1

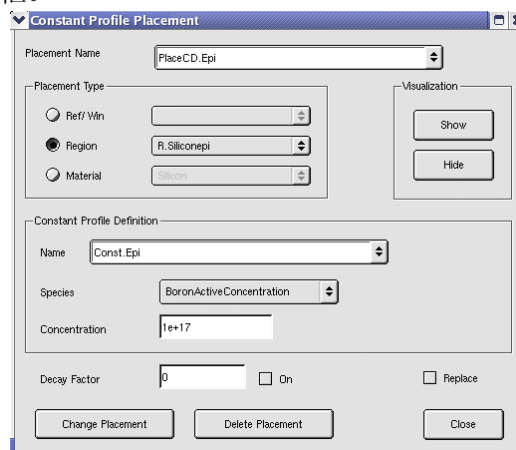


图 2

定义某材料的掺杂浓度为常数，通常包括两步：

第一步是定义一个常数分布，要求完成图 1 中 Constant Profile Definition 组框中设置。

第二步是将定义好的分布与一种材料类型相关联，这可以通过 SDE 的 Add Placement 按钮实现。但是，在关联之前，在图 1 中 Placement Type 组框中的 placement 类型和材料类型都必须选择。

C.2 定义区域的掺杂浓度为常数

定义材料的掺杂浓度为常数时，所有同种类型的材料（可以包括多个区域）均具有相同的掺杂浓度。也可为选择的区域（一个或多个区域）指定掺杂浓度。下面为外延层指定掺杂浓度为 $1 \times 10^{17} \text{cm}^{-3}$ 。

1、Device->Constant Profile Placement, 弹出图 2 的对话框。

注意：通过在 Placement Type 组框中选择 Ref/Win，也可将常数掺杂分布应用到求值窗口。

C.3 定义解析的掺杂分布

在 SDE 中，可以采用高斯和误差函数等定义解析的掺杂分布。除此之外，SDE 允许用户采用自己的函数定义掺杂分布。

通常，解析掺杂分布的指定（placement）分为两步。首先，定义基准；其次，定义掺杂分布的形状。基准用于定义掺杂的横向分布，也可作为峰值位置深度的参考点。

本例将为源/漏区指定两个高斯掺杂分布。源/漏区的目标是指定 P 分布为高斯分布，峰值浓度为 $5 \times 10^{19} \text{cm}^{-3}$ ，结深（junction depth）为 $0.12 \mu\text{m}$ ，横向扩散系数为 0.8。

源/漏延伸区的目标是，指定 As 分布为高斯分布，峰值掺杂浓度为 $5 \times 10^{18} \text{cm}^{-3}$ ，结深为 $0.035 \mu\text{m}$ 。

首先定义基准：

1、Mesh->Define Ref/Eval Window->Line

2、在显示窗口中，点击基准线的第一个点，弹出 Exact Coordinates 对话框

3、指定第一个点为：(-0.8 0)

4、定义第二个点为：(-0.2 0)

在弹出的对话框中，输入名称 BaseLine.Source。

参照下表进行类型定义：

Junctions	Baseline Name	Start Point	End Point
Source	BaseLine.Source	(-0.8 0)	(-0.2 0)
Drain	BaseLine.Drain	(0.2 0)	(0.8 0)
Source extension	BaseLine.SourceExt	(-0.8 0)	(-0.1 0)
Drain extension	BaseLine.DrainExt	(0.1 0)	(0.8 0)

其次定义和关联解析掺杂分布：

1、Device->Analytic Profile Placement，弹出 Analytic Profiles 对话框

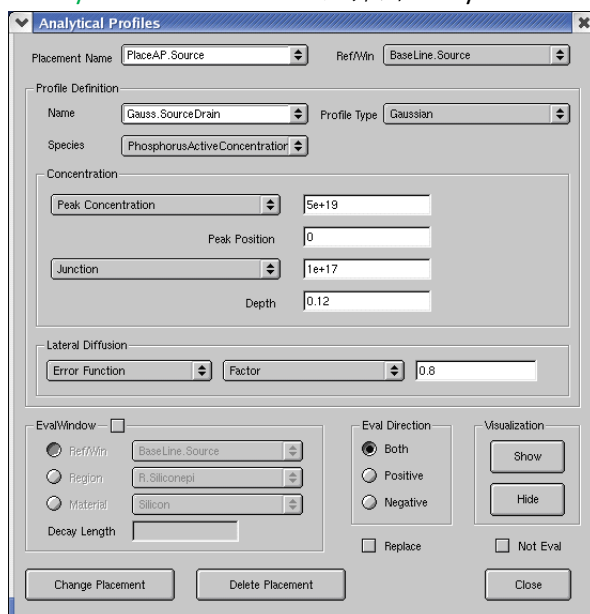


图 3

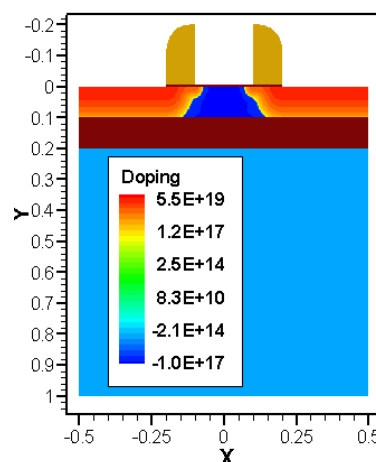


图 4

按照以下两个表进行类似设置。

Junctions	Placement Name	Baseline Name	Profile Name
Source	PlaceAP.Source	BaseLine.Source	Gaussian.SourceDrain
Drain	PlaceAP.Drain	BaseLine.Drain	Gaussian.SourceDrain
Source Extension	PlaceAP.SourceExt	BaseLine.SourceExt	Gaussian.SourceDrainExt
Drain Extension	PlaceAP.DrainExt	BaseLine.DrainExt	Gaussian.SourceDrainExt

掺杂分布名称	峰值浓度	峰值位置	结的浓度	结深	横向扩散系数
Gaussian.SourceDrain	$5 \times 10^{19} \text{cm}^{-3}$	$0 \mu\text{m}$	10^{17}cm^{-3}	$0.12 \mu\text{m}$	0.8

Gaussian.SourceDrainExt	$5 \times 10^{18} \text{ cm}^{-3}$	0 μm	10^{17} cm^{-3}	0.035 μm	0.8
-------------------------	------------------------------------	-----------------	---------------------------	---------------------	-----

Q: 高斯分布中各参数与最终掺杂浓度的具体对应关系？

C.4 保存模型(见图 4)

D. 生成网格（20 分钟）

目标：设置二维器件的网格生成策略

D.1 简介

本节介绍器件结构网格的生成。在数值解算半导体器件的电特性之前，通常需要生成网格。
在 SDE 中，网格的生成过程包括两步。第一部定义网格化策略，包括定义各坐标方向上的最大和最小网格尺寸以及定义网格优化策略；第二步是将定义的策略与特定的对象（通常为材料、器件区域或者用户定义的求值窗口）相关联。
注意：在确定器件性能时，不同的器件区域扮演着不同的角色，因此这些区域的网格策略通常是不同的。

D.2 定义区域的网格化策略

为器件区域定义一个简单的网格化策略。
1、Mesh->Refinement Placement，弹出如下图所示对话框

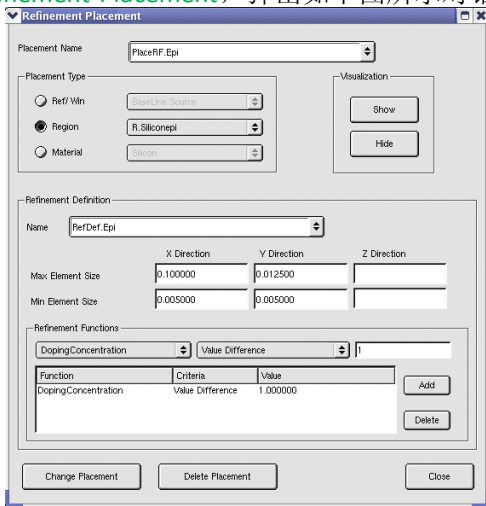


图 1

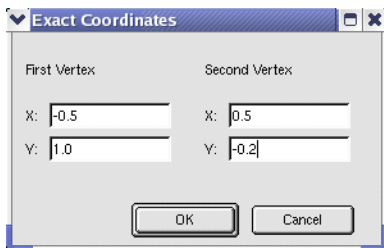


图 2

D.3 定义优化窗口

在 SDE 中，网格化策略可以被限制在选定的区域内。在应用网格化策略之前，需要首先定义优化窗口。
定义一个覆盖整个器件的优化窗口

1、Mesh->Define Ref/Eval Window->Rectangle

2、采用鼠标画一个覆盖整个器件的矩形

3、在弹出的窗口中输入图 2 中所示的值

4、点击 OK 按钮，输入优化名称为 RefWin.all

5、重复以上步骤，定义一个只包括沟道区域的优化窗口。坐标分别为(-0.1 0.0)和(0.1 0.1)，名称为 RefWin.Channel。

D.4 定义优化窗口的网格化策略

在定义好优化窗口之后，设置优化窗口的网格化策略与设置区域的网格化策略是一样的。

唯一的区别是在 Placement Type 中选择 Ref/Eval:

1、Mesh->Refinement Placement, 弹出图 3 所示的对话框

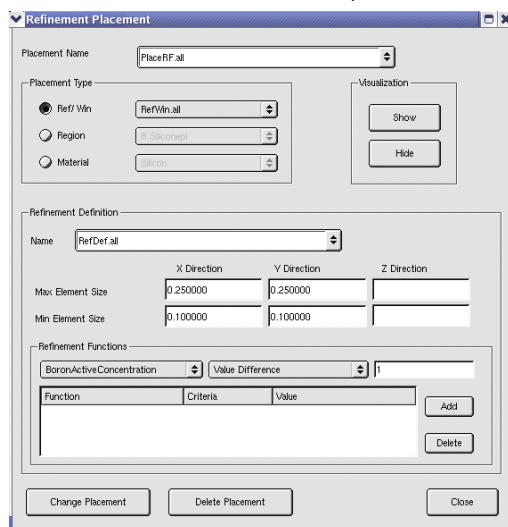


图 3

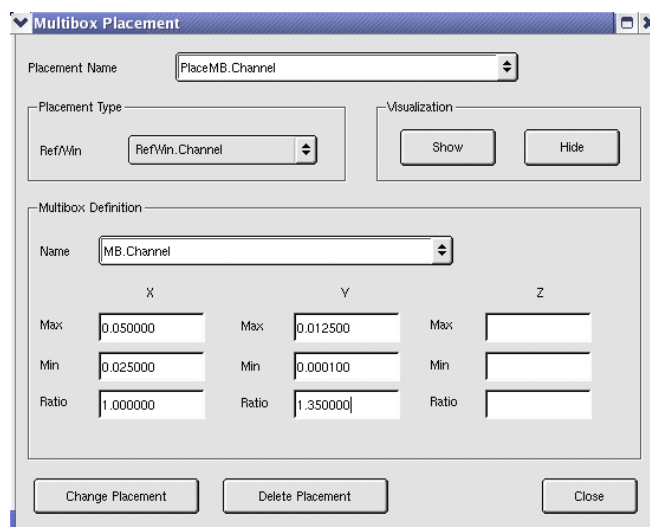


图 4

D.5 在优化窗口中定义一个 Multibox 网格化策略

在某些应用中，需要渐近地改变网格线的密度。例如，在 MOS 晶体管的沟道区，需要设置很小的垂直网格间距，随着向体区的不断延伸，垂直网格间距需要不断变大，这样既可适应反型层的高载流子浓度梯度，又可适当减少 CPU 执行时间。

SDE 支持另一种称为 Multibox 优化策略的网格化策略。

1、Mesh->Multibox Placement, 弹出如图 4 所示的窗口

Ratio 为 1 表示没有网格弛豫。Ratio 等于 1.35 表明：如果最小的间距为 0.1nm，则第二个间距为 0.135nm，第三个间距为 0.182nm，等等。但是，处于其它结构和数值方面的考虑，二叉树/八叉树算法不会严格地产生所需的网格间距。

注意：尽管 ratio 可以为 1-2 之间的值，但内部算法产生的相邻网格之间的间距均为 2。因此，实际看到的网格是一组一倍间距的网格，一组二倍间距的网格，一组四倍间距的网格，等等。Ratio 的符号决定 multibox 的渐近从哪边开始，如果符号为正，表示渐近从左边或者顶部开始，反之，则从右边或者底部开始。

D.6 保存模型

下一步网格化引擎需要用到的两个文件为：**soifet.cmd** 和 **soifet.bnd**。

D.7 为器件结构产生网格

到此为止，为不同的区域或者窗口定义了多种网格化策略。为了生成器件的网格，必须调用网格化引擎。

1、**Mesh->Build Mesh**，弹出如图 5 所示的对话框。

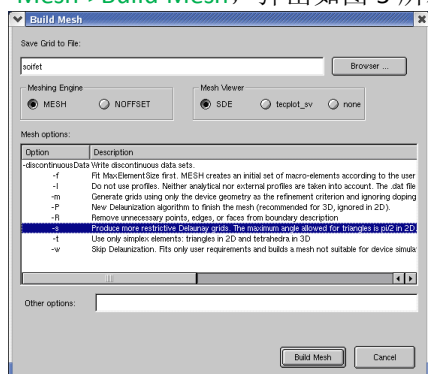


图 5

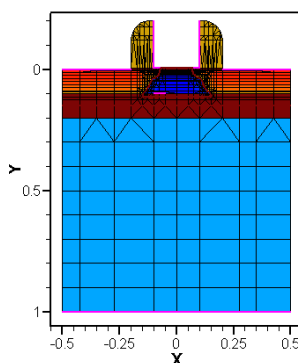


图 6

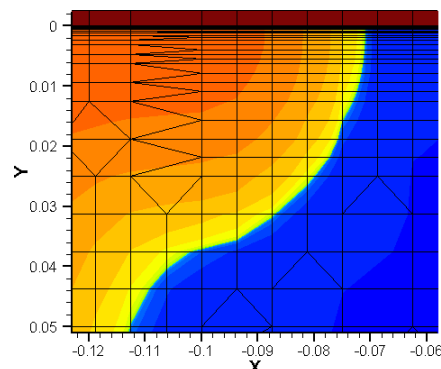


图 7

Mesh 引擎产生两个输出文件 **soifet_msh.grd** 和 **soifet_msh.dat**，这两个文件可作为器件模拟的输入。

图 6 所示为带有掺杂浓度和网格信息的器件结构（采用 TecplotSV 查看）

图 7 所示为源/沟道结区的放大图。

E. 脚本和参数（30 分钟）

目标：在 SDE 输入文件中使用脚本。

E.1 简介

SDE 的脚本语言基于 Scheme，Scheme 是一个类似 LISP 的编程语言，该编程语言与通常使用的编程语言具有很大差别。为了采用 SDE 脚本创建参数化器件，需要一定的与 Scheme 相关的基础知识。

本节介绍最常用的 Scheme 命令。

E.2 Scheme 基础

可以采用两种方式执行 Scheme 命令，一种是在命令行窗口中输入 Scheme 命令，另一种是加载 Scheme 脚本：

1、File->Import->File Type:Scheme Script Files

Scheme 命令均是包含在括号中：

(Scheme command)

；之后的部分均被视为注释

Scheme 命令可以跨多行，一行中可以包括多个 Scheme 命令。

E.3 定义简单的变量和数据类型

整数和浮点数被视为数字。采用下面的方式声明和定义一个数字变量：

```
(define i 3)
```

```
(define pi 3.141593)
```

符号串被包含在双引号中。采用下面的方式声明和定义符号串：

```
(define W "Hello World")
```

声明和定义字符：

```
(define CHAR 'a)
```

更新一个已经存在的变量：

```
(define j 1)
```

```
(set! j (+ j 1))
```

注意：采用 `define` 首次定义一个变量。采用 `set!`更新一个已经存在的变量。

E.5 数学操作符

- `+-*/`
- 三角函数： `sin z`, `cos z`, `tan z`, `asin z`, `acos z`, `atan z`
- 指数等函数： `exp z`, `log z`, `sqrt z`
- 幂次函数： `expt z1 z2` ($z1^{z2}$)
- Rounding: `floor z`, `ceiling z`, `truncate z`, `round z`
- 最大和最小： `max z1 z2 ...`, `min z1 z2 ...`

E.6 数学表达式

定义变量 `j=i+5`

```
(define j (+ i 5))
```

定义变量 `SIN=sin(pi/2)`

```
(define SIN (sin (/ pi 2)))
```

定义变量 `x=x0+R*cos(2*pi*fi/360)`

```
(define x0 1)
```

```
(define R 0.2)
```

```
(define fi 30)
```

```
(define x (+ x0 (* R (cos (/ (* 2 pi fi) 360)))))
```

Noffset3D

Noffset3D 是一个网格生成器，在二维时产生三角形和矩形，在三维时产生四面体。Noffset3D 面向 Sdevice 和 Sprocess 等采用方框法作为离散化方法的模拟器。

A. Getting Started 10 min

A.1 Overview

Noffset3D 是一个网格生成器，在二维时产生三角形和矩形，在三维时产生四面体。其主要思想就是方框法，构建的网格是表面自适应的、各项异性的 mesh 层，这与 Delaunay mesh 相一致。

Noffset3D 可被用于任何使用 mesh 的场合。但是，与 Mesh 不同，Noffset3D 产生的网格极端自由的，网格线通常不与坐标轴平行。

Noffset3D 沿特定的表面生成各向异性的网格层，从而达到与器件的几何结构自适应。也可采用其它的算法，以适应其它数据，例如 pn 结的网格优化。

Noffset3D 也可被看作多种网格化算法的前台。

A.2 Starting Noffset3D

% noffset3d command.cmd boundary.bnd （需要同时指定命令文件和边界文件）

B. Using Noffset3D in 2D 40 min

B.1 Strategies of Meshing

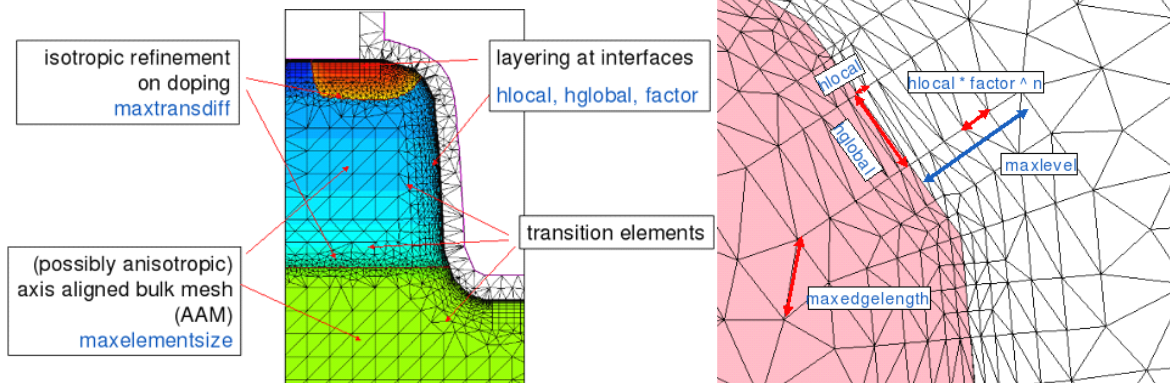
算法的流程一般是“从外到内”，首先生成表面的网格，然后生成第一个 volume，并在该 volume 中填充网格点。这样一个流程的优点如下：

- 边界处的网格比较细；
- 用于常用几何结构时，具有较高的鲁棒性；
- 灵活性。

但是，生成的网格是 unstructured，很难找到优化的参数。

B.2 Main Parameters for Meshing

生成网格时需要四种类型的参数。



- maxtransdiff 根据掺杂浓度对网格进行各向同性的优化。
- maxelementsize 定义最大边长。
- hlocal, hglobal, and factor 主要与边界处的网格层相关。

C. Using Noffset3D in 3D 40 min

C.1 Main Algorithm

Noffset3D 执行多种网格化操作。首先，Noffset3D 采用与方框法兼容的俄 Delaunized 网格。其次，Noffset3D 允许表面自适应的网格和与边界相适应的网格。Noffset3D 生成各向异性和各向同性的网格。所有这些操作均通过算法描述。

本节描述 Noffset3D 包含的算法。

通过 -p 选项可以指定算法。可以通过多个 -p 选项指定多种算法。Noffset3D 在执行网格化操作时保留指定的顺序。

Surface: Triangulates surfaces and boundaries.

Noffset3d: Boundary-conforming meshes.

Refine: Isotropic, unstructured bulk meshes, Mesh-like refinement boxes.

Junctions: p-n junction-conforming meshes.

Field: Incorporates external points.

Deltri: Volume triangulation (delaunization).

Optimize: Improves mesh quality without enforcing Delaunay criterion.

Redcon: Reduces connectivity by adding points.

Sentaurus Device

Sentaurus Device 是一个复杂的半导体器件模拟器，可用于模拟 Si 基和化合物半导体的电、热和光特性。本节介绍 Sentaurus Device 的基本特性和功能。

A. 基础（15 分钟）

A.1 简介

Sentaurus Device 是一个复杂的半导体器件模拟器，可用于模拟 Si 基和化合物半导体的电、热和光特性。Sentaurus Device 可以模拟很宽工作条件范围内的一维、二维和三维几何结构，包括混合模拟。

A.2 输入命令文件

Sentaurus Device 的典型输入命令文件包括多个命令段（或者称为语句块），每个命令段完成相对独立的功能。输入命令文件的默认形式为 `_des.cmd`，例如 `n1_des.cmd`。

注意：命令文件中的关键词是不区分大小写的，且绝大部分有对应的缩略语（译者注：推荐使用全称，以增加可读性）。但是，外部引用，例如文件名和接触名（器件结构文件中定义的接触）是区分大小写的且必须包含在引号内（译者注：推荐使用双引号）。

输入命令文件通常包括以下段：

File

Electrode

Physics

Plot

Math

Solve

File

File 段定义模拟的输入和输出，例如：

```
File {  
    * Input Files  
    Grid = "nmos_msh.tdr"  
    Parameter = "nmos.par"  
    * Output Files  
    Current = "n1_des.plt"  
    Plot = "n1_des.tdr"  
    Output = "n1_des.log"  
}
```

Sentaurus Device 需要一个基本的输入文件定义器件的结构和各个域的值（例如：掺杂浓度）。除此之外，可以定义一个参数文件。

TDR 格式的网格和掺杂文件包含两类信息：器件的几何结构，包括器件的区域和材料，接触的位置，网格点（包括所有离散点的位置，也被称为节点或顶点）；器件的域值（例如：掺杂浓度）。结构文件可以是一维、二维或者三维，通常由 Mesh 引擎产生。默认的扩展名为 .tdr。

注意：当采用 DF-ISE 格式时，Sentaurus Device 需要两个基本的文件定义器件的结构（网格文件的扩展名为 .grd）和域值（如掺杂浓度，扩展名为 .dat）。从 TDR 变为 DF-ISE 格式时，替换 Grid = "nmos_msh.tdr" 为：

Grid = "nmos_msh.grd"

Doping = "nmos_msh.dat"

译者注：不论是哪种格式，都需要经过 Mesh 引擎之后才能被用于器件模拟。两种格式只是表现形式的不同，没有本质区别。推荐在 Mesh 时生成 .tdr 格式。

Plot 文件（TDR 格式）包括所有变量的最终模拟结果，例如在最终偏压点下的电子分布。默认的扩展名为 _des.tdr。

注意：当采用 DF-ISE 格式时，在 File 段用 Plot = "n1_des.dat" 替换 Plot = "n1_des.tdr"。

可选的参数文件用于定义用户指定的模型参数。参数文件中指定的参数值的优先级高于 Sentaurus Device 的内建默认值。参数文件的默认扩展名为 .par。

Sentaurus Device 模拟产生多个输出文件。Current 文件包含电特性输出数据，例如每个接触的电流、电压和电荷。默认的扩展名为 _des.plt。

Output 文件包含 Sentaurus Device 运行时的所有信息，包括采用的物理模型以及相应的参数值。该文件也包含错误信息。默认的扩展名为 _des.log。

图 1 给出了 Sentaurus Device 的典型输入、输出文件流程图。

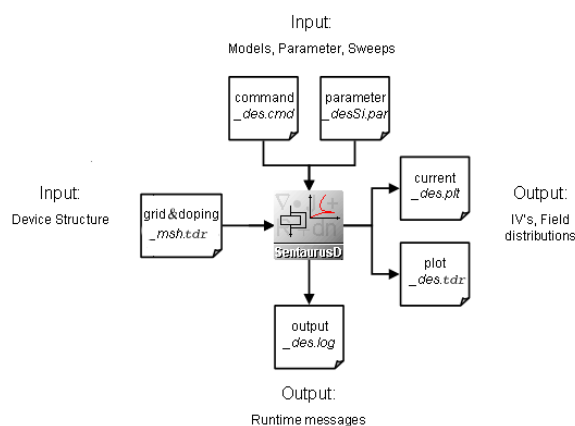


图 1 Sentaurus Device 的典型输入、输出文件流程图

器件的电（或者热）接触及其初始偏压条件在 Electrode 段进行定义。如果一个接触存在特殊的边界条件，可以在此进行定义。

此处定义的每一个电极必须与结构文件中存在的接触名严格对应（包括大小写），只有在 Electrode 段定义过的接触才会被模拟（译者注：在器件结构中存在的接触如果没有在 Electrode 段定义，则不会被模拟，就像在器件结构中没有该接触一样）。

Electrode

Electrode {

```
{ Name="source" Voltage=0.0 }
{ Name="drain" Voltage=0.0 Resistor=100}
{ Name="gate" Voltage=0.0 Barrier=-0.55}
```

...

```
{ Name="base"      Voltage=0.0 Current=0 Barrier=-0.55}
...
{ Name="HEMTgate" Voltage=0.0 Schottky Barrier=0.78}
}
```

注 1：在漏极连接了一个 100 Ohm* μm 的集中电阻。

注 2：如果栅接触被当作理想的接触处理，亦即，不是一个多晶硅区，必须显式定义功函数。通常采用参数 Barrier 定义功函数。

Barrier 参数定义本征 Si 材料和(高掺杂)多晶硅栅之间的功函数差。对于 N+ 多晶硅，功函数差约为 -0.55V，也就是 Si 禁带宽度的一半。（Q：多晶硅的掺杂浓度对功函数差有无影响？有多大的影响？典型的 VDSM 工艺采用的多晶硅栅的掺杂浓度为多少？）

在 base 小段中的 Current 语句将 base 接触定义为一个电流源而不是一个电压源（注：默认值为电压源）。电流源的电流值可以在 Solve 段中 ramp。（Q：有何用处？）

接触 HEMTgate 被声明为一个 Schottky 接触，势垒高度为 0.78V。

Physics

在 Physics 段定义模拟采用的物理模型，例如载流子迁移率模型、禁带变窄模型、载流子的产生和复合模型、碰撞电离模型和栅漏流模型。

```
Physics {
  [list of models]
}
```

或者：

```
Physics (Material="[material name]") {
  [list of models]
}
```

或者：

```
Physics (RegionInterface="[region name]") {
  [list of models]
}
```

没有任何限定词的 Physics 段表示该段中的模型对器件结构的所有区域都是有效的。带有限定词（如：Material="[material name]" 或者 Region="[region name]"）的物理段只对相应的材料或区域有效。类似地，模型也可限定于材料或者是区域的界面。

注意：在 Physics 段中，只声明或者激活了模型本身。可以在参数文件中对模型参数进行修改。

下面给出了典型 NMOSFET 模式采用的 Physics 段。

```
Physics {
  Mobility( DopingDep HighFieldSat Enormal )
  EffectiveIntrinsicDensity( OldSlotboom )
}
```

在上例中，载流子迁移率模型考虑了：

- 高掺杂浓度引起的迁移率下降，DopingDep；
- 高电场下的速度饱和，HighFieldSat；
- 表面散射引起的迁移率下降，Enormal。

Plot

```
Plot {
  [list of variables]
}
```

Plot 段用于定义解算的变量，在模拟结束后保存在 Plot 文件中，或者在 Solve 段采用 Plot 命令也可保存

文件（这样可以在任意的偏置点保存文件）。变量的解算结果可以采用其它工具如 Tecplot SV 查看。解算的变量例子包括掺杂浓度和载流子浓度、电流密度、电场分布。

对于矢量，例如电场强度，在变量后可以附加指示符"/Vector"或者"/V"保存变量的矢量值。例如：

```
Plot {
  eCurrent/v
}
```

下面是一个典型的 Plot 段：

```
Plot {
  eDensity hDensity eCurrent hCurrent
  Potential SpaceCharge ElectricField
  eMobility hMobility eVelocity hVelocity
  Doping DonorConcentration AcceptorConcentration
}
```

Math

Math 段用于控制模拟时的数值计算。下面的 Math 段是一个典型的 Math 段，可被用于指南：

```
Math {
  Extrapolate          *off by default
  RelErrControl        *on by default
  Iterations=20        *default = 50
  Notdamped=100        *default = 1000
}
```

Extrapolate: 在偏置 ramp 时，特定偏置点解算的初始猜测通过对前两步结果的插值获得。

RelErrControl: 激活改进的误差控制策略。

Iteration=20: 牛顿迭代的最大次数为 20。如果在指定的迭代次数之后仍不收敛，则减小偏置的步进值或者时间扫描的步进值。

Notdamped=100: 前 100 次牛顿迭代不采用衰减算法。在大多数情况下，不推荐使用衰减（damping）。

Solve

在 Solve 段的偏压扫描为一个标准针对 MOSFET 的 IdVg 扫描。

```
Solve {
  *- Build-up of initial solution:
  Coupled(Iterations=100){ Poisson }
  Coupled{ Poisson Electron Hole }
  *- Bias drain to target bias
  Quasistationary(
    InitialStep=0.01 Increment=1.35
    MinStep=1e-5 MaxStep=0.2
    Goal{ Name="drain" Voltage= 0.05 }
  ){ Coupled{ Poisson Electron Hole } }
  *- Gate voltage sweep
  Quasistationary(
    InitialStep=1e-3 Increment=1.35
    MinStep=1e-5 MaxStep=0.05
    Goal{ Name="gate" Voltage= 1.5 }
  ){ Coupled{ Poisson Electron Hole } }
```

}

首先解泊松方程得到初始解。初始偏压条件来自 Electrode 段。为了保证收敛，这一步允许 100 次牛顿迭代。

下一步，在一个牛顿解算器中自相容地（self-consistently）解算泊松方程和载流子连续性方程。

第一条 Quasistationary 语句将漏极电压从初始电压 ramp up 到 0.05V。实际上，Quasistationary 命令将一个内部变量 t 从 0 ramp 到 1，采用如下公式计算每一步的偏压：

$$V(t) = V_{\text{initial}} + t * (V_{\text{goal}} - V_{\text{initial}})$$

在 Quasistationary 语句中指定的所有控制参数应用于变量 t 的 ramp。InitialStep 设置变量 t 的初始步进值（Initial Step），本例中为 0.01。相应的初始步进电压为：

$$V_{\text{initial}} + \text{InitialStep} * (V_{\text{goal}} - V_{\text{initial}}) = 0.0005 \text{ V, if } V_{\text{initial}} = 0$$

根据前一步的结果，步进值会自动增加或者减小。

- 如果前一步的偏压成功收敛，下一步的步进值增加 Increment 倍（Increment 值默认为 2，本例中设为 1.35）
- 如果前一步的偏压没有收敛，下一步的步进值减小 Decrement 倍（默认值为 2）
- 如果步进值缩小到小于 Minstep，扫描失败（abort），得出模拟不收敛的结论。
- 步进值的最大值为 MaxStep。

类似地，第二条 Quasistationary 语句将栅压 ramp 到 1.5V。在栅压 ramp 的过程中，漏极电压依然为 0.05V。

A.3 参数文件

在 Sentaurus Device 中，物理模型在 Physics 段定义。如果模型参数与默认值不同，则采用参数文件的方式在 File 段加载。

产生一个默认参数文件的副本，在命令行下键入：

B2500-4%`sdevice -P`

这一命令产生 sdevice.par，包含当前 Sentaurus Device 版本使用的所有默认模型参数。

为了改变模型的参数，例如 SRH 复合模型采用的参数：

- 1、产生一个默认参数文件的副本，例如 sdevice.par
- 2、找到文件中与 SRH 复合相关的部分。

Scharfetter * relation and trap level for SRH recombination:

```
{ * tau = taumin + ( taumax - taumin ) / ( 1 + ( N/Nref )^gamma)
  * tau(T) = tau * ( (T/300)^Talpha )          (TempDep)
  * tau(T) = tau * exp( Tcoeff * ((T/300)-1) ) (ExpTempDep)
    taumin   = 0.0000e+00 , 0.0000e+00      # [s]
    taumax   = 1.0000e-05 , 3.0000e-06      # [s]
    Nref      = 1.0000e+16 , 1.0000e+16     # [cm^(-3)]
    gamma     = 1 , 1                        # [1]
    Talpha    = -1.5000e+00 , -1.5000e+00   # [1]
    Tcoeff    = 2.55 , 2.55                  # [1]
    Etrap     = 0.0000e+00                    # [eV]
}
```

注意：除参数 Etrap 之外，每个参数均有两个值：一个对应电子，一个对应空穴。Etrap 定义 defect level 和 intrinsic level 之间的差别。

- 3、生成一个新文件，例如 newSRHparameters.par。将 sdevice.par 文件中的 SRH 部分 copy 到新文件中。修改相关的参数。

4、在 File 段引用新文件。

B. 载流子传输模型（45 分钟）

目标：了解各种载流子传输模型及其应用

B.1 简介

本节对各种载流子传输模型进行简介，以及漂移-扩散、热力学、流体力学、Monte Carlo 传输和量子传输（基于密度梯度模型）。这些模型应用与不同的场合，根据应用需求灵活选择。

为了理解其应用，每种模型都给出了一个 Sentaurus Device 的输入命令文件例子。

本节采用的示例文件为 [TransportModels.gzp](#)。用户需要将 TransportModels.gzp 文件 copy 到 \$STDB 目录下，然后采用 [Project->Import\(tar\)](#) 导入工程。

节点 3-6 分别对应漂移扩散模型、流体力学模型、热力学模型和量子力学模型。

B.2 漂移-扩散传输

采用漂移-扩散传输模型时，在特定的边界条件下，针对指定的器件区域，自相容地（self-consistently）解算耦合的泊松和载流子（电子 and/or 空穴）连续性方程。这一模型适合大尺寸器件（0.25 及以上工艺），并且电流密度较小的情况。

对于漏极偏压较小的 MOSFET，通常解算泊松和一种载流子的连续性方程（例如：对于 NMOS 器件，解算电子连续性方程）便已足够。

在高漏极偏压条件下，产生-复合过程（例如热的 SRH 产生、雪崩产生、带带隧穿）将变得重要。此时，需要解算两种载流子连续性方程以增加模拟精度。

Sentaurus Device 的输入文件 DD_des.cmd 采用基本的漂移-扩散传输模型模拟 N 管的 IdVg 性能。

文件的主要部分前面已有讨论。本节只讨论与本例密切相关的或者是新的部分（红色部分）。

Physics

```
Physics{
  eQCvanDort
  EffectiveIntrinsicDensity( OldSlotboom )
  Mobility(
    DopingDep
    eHighFieldsaturation( GradQuasiFermi )
    hHighFieldsaturation( GradQuasiFermi )
    Enormal
  )
  Recombination(
    SRH( DopingDep )
  )
}
```

对于深亚微米 MOSFET 而言，沟道区的载流子量子化（归因于垂直于半导体/绝缘体界面的高电场）变得重要。唯像的 van Dort 量子修正模型考虑了这一效应（略微增加了计算复杂度）。Physics 段的 eQCvanDort 语句激活了 **电子的唯像 van Dort 量子修正模型**。

EffectiveIntrinsicDensity 语句考虑了高掺杂区的禁带变窄效应。这一模型直接影响 Si 中本征载流子浓度的计算。

Mobility 语句定义了载流子迁移率模型，在本例中，包括：高掺杂浓度引起的迁移率下降、高横向电场区的速度饱和、半导体/绝缘体界面附近高纵向电场引起的迁移率下降。**GradQuasiFermi** 语句指定饱和模型的驱动力为准费米势的梯度。

Solve

```
Solve {
  *- Build-up of initial solution:
  NewCurrentFile="init"
  Coupled(Iterations=100){ Poisson }
  Coupled{ Poisson Electron }
  *- Bias drain to target bias
  Quasistationary(
    InitialStep=0.01 Increment=1.35
    MinStep=1e-5 MaxStep=0.2
    Goal{ Name="drain" Voltage= 1.1 }
  ){ Coupled{ Poisson Electron } }
  *- gate voltage sweep
  NewCurrentFile=""
  Quasistationary(
    InitialStep=1e-3 Increment=1.35
    MinStep=1e-5 MaxStep=0.05
    Goal{ Name="gate" Voltage= 1.1 }
  ){ Coupled{ Poisson Electron }
    CurrentPlot(Time=(Range=(0 1) Intervals=20))
  }
}
```

在 Solve 段有两个 Quasistationary 语句。第一条将漏极电压 ramp 到 0.05V。类似地，第二条将栅压 ramp 到 1.5V。

NewCurrentFile 语句用于分离两个 sweep 产生的电压/电流输出文件（*_des.plt）

第一条 **NewCurrentFile**="init" 语句将第一个 sweep 得到的电压/电流输出保存在 init_n3_des.plt 文件中（注：n3_des.plt 是 File 段指定的输出文件名）。

第二条 **NewCurrentFile**="" 语句将第二个 sweep 得到的电压/电流输出保存在 n3_des.plt 文件中。

本例中只解算了泊松和电子连续性方程。

CurrentPlot 语句保证输出文件中为 21 个等间距的 I-V 点。但是，为了保证收敛性，Sentaurus Device 在内部计算时可能多于 21 个点（最终保存在输出文件中的点只有 21 个）。

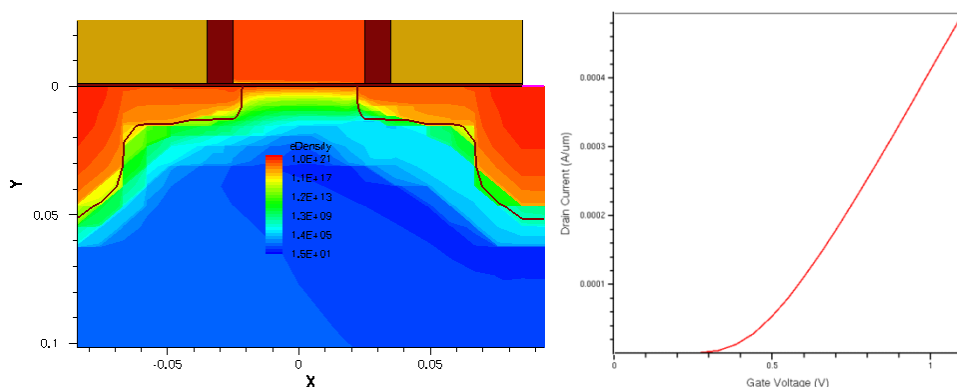


图 1 $V_{gs}=1.1V, V_{ds}=1.1V$ 时沟道区的电子浓度分布 图 2 $V_{ds}=1.1V$ 时，漏极电流随栅压的变化

B.3 热力学传输

热力学模型在漂移-扩散模型的基础上考虑了电致热效应（假设带电的载流子与晶格处于热平衡）。实际上，除了解算泊松和载流子连续性方程之外，还解算了晶格温度方程（热流方程）。

在高电流密度条件下，会发生严重的自热效应，此时需要使用热力学传输方程。例如：功率器件、高栅压和高漏极电压的 MOSFET 以及开环三极管的模拟。

理由在于，高电流在器件区产生焦耳热，这将显著增加晶格温度。由于模拟时的许多模型（如：载流子迁移率模型、SRH 产生复合模型、雪崩产生模型）均与晶格温度有关，为了提高模拟的精度，解算热流（heat flow）是必需的。

热力学传输模型可以单独使用或者与流体力学模型结合使用。对于后一种情况，根据晶格温度分别计算电子和空穴温度。

Thermo_des.cmd 中，在高栅压和高漏极电压条件下模拟 NMOS 管的性能。

Thermode

Thermode{ { Name="substrate" Temperature=300 SurfaceResistance=5e-4 } }

Thermode 语句定义一个热接触。在采用热力学模型时，至少需要定义一个热接触。

热接触可以与电极相同或者定义一个单独的热接触。对于每一个热接触，必须声明一个对应的温度。热接触可以附加一个热阻，其阻值采用关键词 SurfaceResistance 指定，热阻的单位为 $\text{cm}^3\text{K/W}$ 。

没有热阻时，接触的温度一直保持为声明的温度。可以采用非零的热阻仿真器件和外界环境的热交换。

Physics

```
Physics{
  Thermodynamic
  eQCvanDort
  EffectiveIntrinsicDensity( OldSlotboom )
  Mobility(
    DopingDep
    eHighFieldsaturation( GradQuasiFermi )
    hHighFieldsaturation( GradQuasiFermi )
    Enormal
  )
  Recombination(
    SRH( DopingDep )
  )
}
```



```
}
```

Thermodynamic 语句使得在计算器件的电流密度时，将温度梯度也作为一个驱动力（driving force）。通常情况下，温度梯度引起的电流密度与其它部分（例如漂移-扩散部分）比较起来比较小。

Plot

```
Plot{
```

```
*--Density and Currents, etc
```

```
    eDensity hDensity
```

```
    TotalCurrent/Vector eCurrent/Vector hCurrent/Vector
```

```
    eMobility hMobility
```

```
    eVelocity hVelocity
```

```
    eQuasiFermi hQuasiFermi
```

```
*--Temperature
```

```
    eTemperature Temperature * hTemperature
```

```
*--Fields and charges
```

```
    ElectricField/Vector Potential SpaceCharge
```

```
*--Doping Profiles
```

```
    Doping DonorConcentration AcceptorConcentration
```

```
*--Generation/Recombination
```

```
    SRH Band2Band * Auger
```

```
    AvalancheGeneration eAvalancheGeneration hAvalancheGeneration
```

```
*--Driving forces
```

```
    eGradQuasiFermi/Vector hGradQuasiFermi/Vector
```

```
    eParallel hParallel eENormal hENormal
```

```
*--Band structure/Composition
```

```
    BandGap
```

```
    BandGapNarrowing
```

```
    Affinity
```

```
    ConductionBand ValenceBand
```

```
    eQuantumPotential
```

```
}
```

在输出数据文件中包含温度分布。

Solve

```
Solve {
```

```
    *- Build-up of initial solution:
```

```
    NewCurrentFile="init"
```

```
    Coupled(Iterations=100){ Poisson }
```

```
    Coupled{ Poisson Electron Hole Temperature }
```

```
    *- Bias gate to target bias
```

```
    Quasistationary(
```

```
        InitialStep=0.01 Increment=1.35
```

```
        MinStep=1e-5 MaxStep=0.2
```

```
        Goal{ Name="gate" Voltage= 1.1 }
```

```
    ){ Coupled{ Poisson Electron Hole Temperature } }
```

```
    *- drain voltage sweep
```

```
    NewCurrentFile=""
```

```

Quasistationary(
  InitialStep=1e-3 Increment=1.35
  MinStep=1e-5 MaxStep=0.05
  Goal{ Name="drain" Voltage= 1.1 }
){ Coupled{ Poisson Electron Hole Temperature }
  CurrentPlot(Time=(Range=(0 1) Intervals=20))
}
}

```

Coupled 命令中的关键词 Temperature 表示在解算时包含晶格温度。

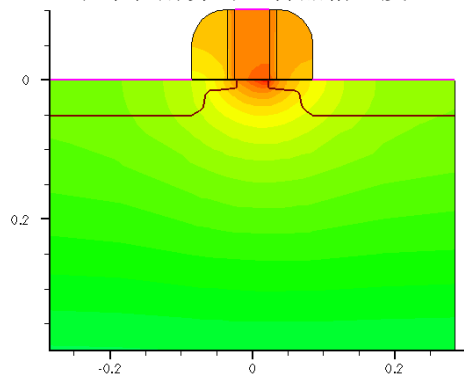


图 3 $V_{gs}=V_{ds}=1.1V$ 时 MOSFET 的温度分布。（峰值温度为 370K）

Q: 热力学模型对于抗辐照有无用处？

B.4 流体力学传输

流体力学模型除了解算泊松方程和载流子连续性方程外，还解算载流子温度和热流方程。

对于 0.18 μm 及以下的深亚微米器件和异质结器件、部分耗尽浮体 SOI 器件，流体力学模型尤其有用。

流体力学模型允许模拟时包含非局部效应（nonlocal effects），例如速度过冲。

流体力学模型的另一个重要应用是模拟器件的击穿。采用流体力学模型避免了过早的击穿（注：在漂移扩散模型中，由于局部电场假设，击穿要比流体力学模型发生得早，实际上不应当那么早击穿）。

流体力学模型的其它应用包括衬底电流的模拟。在这些应用中，通常只有一种载流子的加热效应比较明显，因此，在解算载流子温度方程时只需解算一种类型的载流子。

流体力学模型可以与热力学模型结合使用。

HD_des.cmd 采用流体力学模型模拟 NMOS 管的 IdVd 曲线。

Physics

```

Physics{
  Hydrodynamic(eTemperature)
  eQCvanDort
  EffectiveIntrinsicDensity( OldSlotboom )
  Mobility(
    DopingDep
    eHighFieldsaturation( CarrierTempDrive )
    hHighFieldsaturation( GradQuasiFermi )
    Enormal
  )
  Recombination(

```

```

        SRH( DopingDep )
    )
}

```

Hydrodynamic 语句选择流体力学模型。标识符 **eTemperature** 或者 **hTemperature** 限定哪种载流子使用流体力学传输。如果没有指定标识符，则电子和空穴均采用流体力学模型。

在本例中，只有电子采用流体力学模型，空穴采用漂移-扩散模型。

eHighFieldsaturation 语句中的标识符 **CarrierTempDrive** 表示在高场饱和迁移率下降模型中考虑载流子温度的影响。对于空穴，没有使用该标识符，因为空穴使用的是漂移-扩散模型。

Plot

```

Plot{
*--Density and Currents, etc
    eDensity hDensity
    TotalCurrent/Vector eCurrent/Vector hCurrent/Vector
    eMobility hMobility
    eVelocity hVelocity
    eQuasiFermi hQuasiFermi
*--Temperature
    eTemperature Temperature * hTemperature
*--Fields and charges
    ElectricField/Vector Potential SpaceCharge
*--Doping Profiles
    Doping DonorConcentration AcceptorConcentration
*--Generation/Recombination
    SRH Band2Band * Auger
    AvalancheGeneration eAvalancheGeneration hAvalancheGeneration
*--Driving forces
    eGradQuasiFermi/Vector hGradQuasiFermi/Vector
    eEparallel hEparallel eENormal hENormal
*--Band structure/Composition
    BandGap
    BandGapNarrowing
    Affinity
    ConductionBand ValenceBand
    eQuantumPotential
}

```

在输出文件中包含温度分布。

Solve

```

Solve {
    *- Build-up of initial solution:
    NewCurrentFile="init"
    Coupled(iterations=100){ Poisson }
    Coupled{ Poisson Electron Hole eTemperature }
    *- Bias gate to target bias
    Quasistationary(
        InitialStep=0.01 Increment=1.35
    )
}

```

```

MinStep=1e-5 MaxStep=0.2
Goal{ Name="gate" Voltage= 1.1 }
){ Coupled{ Poisson Electron Hole eTemperature }}
*- drain voltage sweep
NewCurrentFile=""
Quasistationary(
  InitialStep=1e-3 Increment=1.35
  MinStep=1e-5 MaxStep=0.05
  Goal{ Name="drain" Voltage= 1.1 }
){ Coupled{ Poisson Electron Hole eTemperature }
  CurrentPlot(Time=(Range=(0 1) Intervals=20))
}
}

```

为了在计算结果中包含电子温度，Coupled 语句中必须包括关键词 eTemperature。

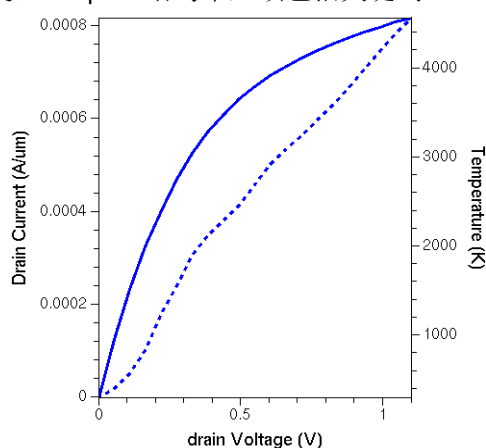


图 4 Vgs=1.1V 时漏极电流对漏极电压的变化（实线）。最大的电子温度（虚线）。

B.5 密度梯度传输

密度梯度模型自相容地与泊松方程和载流子连续性方程解算量子势方程（quantum potential equations）。在经典的器件模拟中通过引入量子势考虑量子效应。在采用密度梯度传输近似时，量子势是载流子浓度和浓度梯度的函数。

浓度梯度传输模型主要用于模拟纳米尺度的器件，例如全耗尽 SOI、双栅 SOI 以及 FinFET 结构。

在体硅 MOSFET 器件中，尺度量子化主要发生在反型层中，简单的唯像 van Dort 模型也能给出很好的结果。在先进的 SOI 工艺下，其它约束（例如超薄层）也有可能引起尺度量子化（size quantization），浓度梯度传输模型将产生更好的结果。

浓度梯度传输模型已经被成功应用于求解二维问题。在三维器件结构中，在与流体力学传输结合之后，浓度梯度模型将更有用。此时，分别求解电子和空穴的温度。

DG_des.cmd 采用浓度梯度模型模拟 N 管的 IdVg 性能。

Physics

```

Physics{
  eQuantumPotential
  EffectiveIntrinsicDensity( OldSlotboom )
  Mobility(

```

```

        DopingDep
        eHighFieldsaturation( GradQuasiFermi )
        hHighFieldsaturation( GradQuasiFermi )
        Enormal
    )
    Recombination(
        SRH( DopingDep )
    )
}

```

关键词 **eQuantumPotential** 表示求解电子的量子势。

Plot

```

Plot{
*--Density and Currents, etc
    eDensity hDensity
    TotalCurrent/Vector eCurrent/Vector hCurrent/Vector
    eMobility hMobility
    eVelocity hVelocity
    eQuasiFermi hQuasiFermi
*--Temperature
    eTemperature Temperature * hTemperature
*--Fields and charges
    ElectricField/Vector Potential SpaceCharge
*--Doping Profiles
    Doping DonorConcentration AcceptorConcentration
*--Generation/Recombination
    SRH Band2Band * Auger
    AvalancheGeneration eAvalancheGeneration hAvalancheGeneration
*--Driving forces
    eGradQuasiFermi/Vector hGradQuasiFermi/Vector
    eParallel hParallel eENormal hENormal
*--Band structure/Composition
    BandGap
    BandGapNarrowing
    Affinity
    ConductionBand ValenceBand
    eQuantumPotential
}

```

eQuantumPotential 语句使得输出文件中包含电子的量子势分布。

为了观察总的有效能带结构，在 **Plot** 文件中添加导带分布和电子量子势。

Solve

```

Solve {
    *- Build-up of initial solution:
    NewCurrentFile="init"
    Coupled(Iterations=100){ Poisson eQuantumPotential }
    Coupled{ Poisson Electron Hole eQuantumPotential }
}

```

```

*- Bias drain to target bias
Quasistationary(
  InitialStep=0.01 Increment=1.35
  MinStep=1e-5 MaxStep=0.2
  Goal{ Name="drain" Voltage= 1.1 }
){ Coupled{ Poisson Electron Hole eQuantumPotential }}
*- Gate voltage sweep
NewCurrentFile=""
Quasistationary(
  InitialStep=1e-3 Increment=1.35
  MinStep=1e-5 MaxStep=0.05
  Goal{ Name="gate" Voltage= 1.1 }
){ Coupled{ Poisson Electron Hole eQuantumPotential }
  CurrentPlot(Time=(Range=(0 1) Intervals=20))
}
}

```

为了解算电子的量子势，在 Coupled 语句中包含了关键词 eQuantumPotential。

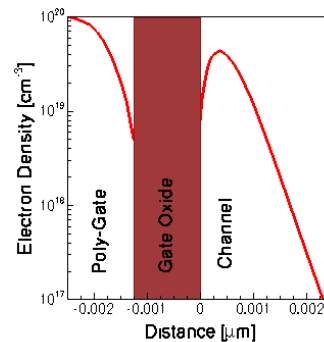


图 5 $V_{gs}=V_{ds}=1.1V$ 时，沟道中心处的电子浓度分布。量子力学的尺度量子化使得电子浓度的峰值离开 Si/SiO₂ 界面约 0.5nm。

B.6 Monte Carlo 传输

表面和沟道方向对 PFinFET 的影响

Monte Carlo 的模拟包括两个主要的步骤。首先运行一个漂移-扩散模拟。其次，以此为基础，进行 Monte Carlo 模拟。漂移-扩散模拟产生的电场被用于初始 frozen-field 模拟（Monte Carlo 传输模拟和泊松方程解算之间的迭代存在一个所谓的初始 frozen-field 模拟）。

Monte Carlo 模拟器，亦即 Sentaurus SPARTA（Single-PARTicle Approach），在整个 Monte Carlo 窗口中对浓度分布进行积分，只模拟积分较高的载流子类型（比如电子的浓度积分较高，则只模拟电子）。另外，Sentaurus SPARTA 假设漂移-扩散模拟能够正确预测主要载流子的积分浓度。因此，与漂移扩散模拟相比，Monte Carlo 模拟只改变了静电势和浓度分布的形状，但并不改变 Monte Carlo 窗口中的总电荷。

模拟的二维几何结构为一个 PFinFET 器件。图 6 为 wafer 的顶视图以及掺杂浓度。沟道区没有掺杂。源漏区掺杂为 $1 \times 10^{20} \text{cm}^{-3}$ 。掺杂的梯度为 5nm/decade。栅长和 fin 宽分别为 50nm 和 34.4nm。

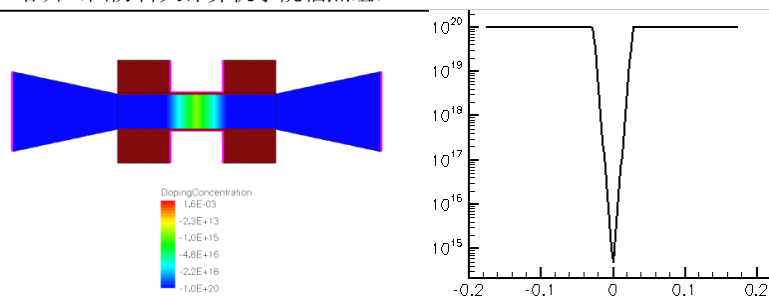


图 6 FinFET 的顶视图和掺杂分布

注意，由于沟道区没有掺杂，量子力学的阈值电压漂移很小，可以忽略。因此，本例中没有考虑量子修正。

本例使用的 SWB 工程为 pFinFET.gzp。

首先进行经典的漂移-扩散模拟，产生电场。模拟结束后，产生一个 n3_des.sav 文件，对应漏极电压为 -1.2V。这一解用于开始 Monte Carlo 模拟（采用命令 SRARTA_des.cmd）。下面简要介绍 SRARTA_des.cmd 文件中的各个段。

File

```
File{
  Grid      = "@tdr@"
  Output    = "@log@"
  Parameter = "@parameter@"
  Load     = "n@node|-2@_des.sav"
  MonteCarloOut = "n@node@_mc"
}
```

在模拟开始时加载文件 n@node|-2@_des.sav 作为初始条件。关键词 MonteCarloOut 决定 Monte Carlo 输出文件的前缀。

Monte Carlo

```
MonteCarlo {
  WithSpartaSimulation
  Normal2OxideDirection = (@surface@)
  ChannelDirection = (@channel@)
  SurfScattRatio = 0.85
  CurrentErrorBar = 2.0
  MinCurrentComput = 5
  DrainContact = 2      # No. of drain contact in .tdr (count from 0,tdx view)
  SelfConsistent(FrozenQF)
  Window = Rectangle[(-0.5, -0.1)
                    (0.5, 0.1)]

  FinalTime = 1.5e-06
  Plot {Range=(0,50e-06) intervals = 500}
}
```

关键词 WithSpartaSimulation 指定采用 Sentaurus SPARTA 作为 Monte Carlo 模拟器。

Normal2OxideDirection = (@surface@) 语句指定 PFinFET 的表面方向是参数化的。在本工程中，定义了两个方向：(0 1 1)和(0 0 1)。

ChannelDirection = (@channel@) 语句指定 PFinFET 的沟道方向是参数化的。在表面方向为(0 1 1)时，定义了两个沟道方向：<0 1 -1>,<1 0 0>,<2 1 -1>。在表面方向为(0 0 1)是，定义了两个沟道方向：<1 0 0> and <-1 1 0>。

在 Sentaurus SPARTA 中，采用镜面反射和漫散射（diffusive scattering，不知翻译得是否正确）模拟表面散射。关键词 SurfScattRatio = 0.85 指定镜面反射的几率。

关键词 CurrentErrorBar = 2.0 指定漏极电流的相对误差（百分数），当相当误差小于该值时，停止模拟。MinCurrentComput = 5 指定达到稳态之后的最小迭代次数，不论漏极电流的相对误差是否小于

CurrentErrorBar。

关键词 SelfConsistent(FrozenQF)定义 Monte Carlo 模拟为 self-consistent。FrozenQF 定义在泊松方程求解过程中冻结的参数。处于稳定性的考虑，必须为准费米势(QF)。

Window = Rectangle[(-0.5, -0.1) (0.5, 0.1)]指定对完全位于矩形(-0.5, -0.1) (0.5, 0.1)内的元素进行 Monte Carlo 模拟。

FinalTime = 1.5e-06 指定达到稳态之后的模拟时间。在 FinalTime 之后进行累积平均值的收集。

Plot {Range=(0,50e-06) intervals = 500}指定经过多少时间之后进行泊松方程的求解（此句翻译不一定正确）。Range=(0,50e-06)指定范围，起点和终点均为浮点数，分别定义起始时间和最大模拟时间。在 Sentaurus SPARTA 中，起点必须为 0。

Math

Math {

Method = blocked

Submethod = pardiso

Digits = 6

Extrapolate

Derivatives

Iterations = 20

NewDiscretization

Notdamped = 50

ConstRefPot

}

Solve

Solve {

coupled {poisson electron}

montecarlo

}

关键词 montecarlo 可以在 Solve 段的任何位置指定，类似一个偏微分方程。montecarlo 不能与任何其它的偏微分方程耦合（coupled）。但是，可被用于亚稳态模拟。

Monte Carlo 模拟的结果示于图 7。纵坐标为开态电流（on-state current），横坐标为沟道方向和表面方向。图 7 表明，沟道方向<1 0 0>和表面方向(0 1 1)的组合对于 PFinFET 来说是有利的。这是两个物理效应的结果。一方面，镜面散射决定于能量守恒和平行波矢量（parallel wave vector）。相对于(0 1 1)方向的界面，能带结构的方向使得具有一个更高的低场迁移率。另一方面，归因于<1 0 0>方向的驱动电场，空穴的高场传输得到加强。

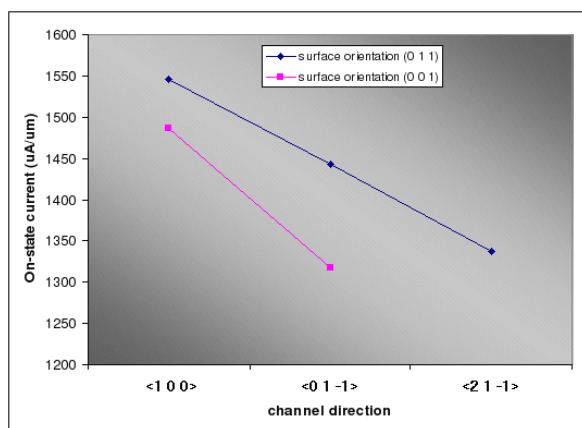


图 7 开态电流随沟道方向和表面方向的变化

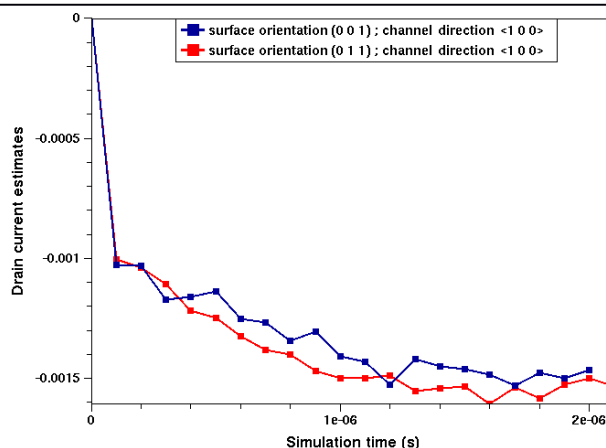


图 8 漏极电流随模拟时间的变化, 沟道方向均为 <100>, 表面方向分别为 (0 0 1) 和 (0 1 1)。在达到稳态之后, 电流开始在平均值附近上下波动。

电流的模拟结果文件为 `mc.sparta_time.plt` 和 `mc.sparta_average.plt`。这两个文件包含接触处的电流的估算值（由关键词 `DrainContact` 指定），在 `.plt` 文件中称为 `MCdrain`，以及在整个 Monte Carlo 窗口内碰撞电离率的积分，称为 `MCsubstrate`。在 `mc.sparta_time.plt` 中，电流的估算值（对应于一次迭代间隔）被储存为模拟时间的函数。

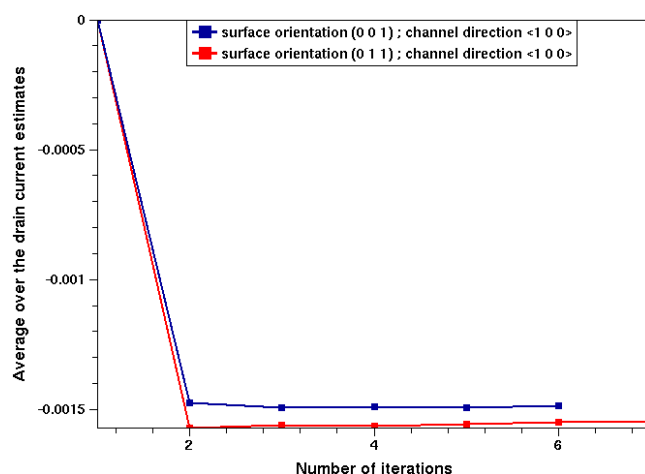


图 9 漏极电流估算期间的累积平均值随迭代次数的变化。

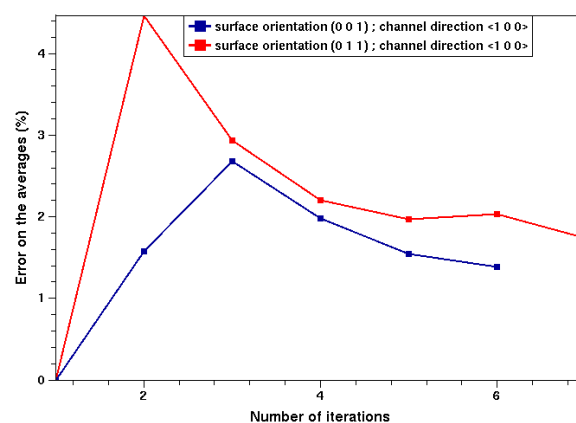


图 10 平均值的相对误差随迭代次数的变化。

除此之外，从 `mc.sparta_average.plt` 文件中可以提取这些平均值的相对误差。这些误差可以作为停止模拟（在达到稳态之后开始收集统计数据）的合理判据。

注意：可以采用 `inspect` 查看 `mc.sparta_time.plt` 和 `mc.sparta_average.plt` 文件。

Q: Monte Carlo 模拟对于辐射领域的意义？

译者注：总的来说，这一小节没看太懂。

C. 混合模式模拟（45 分钟）

C.1 简介

对于混合模拟，Sdevice 的命令文件不仅要包括电路中每个器件的定义，还要包括一个网表用于指定器件之间的连接关系。因此，命令文件中包含额外的节，如 System。除此之外，Solve 节可以指定模拟 System 的一部分或者全部。

混合模拟的例子包括环振的模拟、SRAM 单元的 SEU、包含集中元件和外部寄生参数的 RF 放大器的分析。

C.2 混合模式模拟

本节采用混合模式模拟反相器的电压转换特性（Voltage Transfer Characteristic，VTC）

```
System {
  Vsource_pset vdd (dd 0) { dc = 0.0 }
  Vsource_pset vin (in 0) { dc = 0.0 }

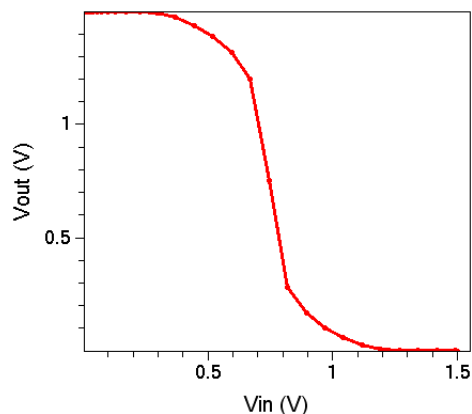
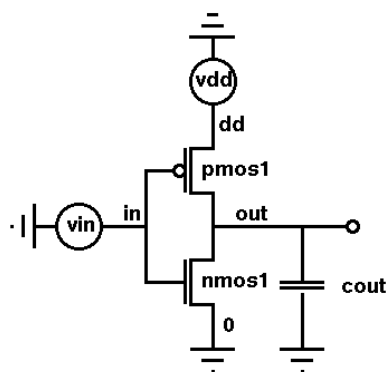
  NMOS nmos1 ( "source"=0 "drain"=out "gate"=in "substrate"=0 )
  PMOS pmos1 ( "source"=dd "drain"=out "gate"=in "substrate"=dd )
  Capacitor_pset cout ( out 0 ){ capacitance = 3e-14 }

  Plot "n4_sys_des.plt" (time() v(in) v(out)
                        i(nmos1,out) i(pmos1,out) i(cout,out) )
}
```

在 System 节中，不仅要定义电路元件，还需要定义元件之间的连接关系。

Vsource_pset 指定了一个电压源的集约模型（Sdevice 的内建模型）。该模型在此处用于定义两个电压源 Vdd 和 Vin。

System 节的 Plot 语句用于记录节点的电压和电流。记录电压时，指定节点名；记录电流时，需要同时指定元件名和电流流过的节点名。



```
Solve{
  NewCurrentFile="init"
  Coupled(iterations=100){ Poisson }
  Coupled{ Poisson Electron Hole Contact Circuit }
```

```

Quasistationary(
    InitialStep=1e-3 Increment=1.35
    MinStep=1e-5 MaxStep=0.05
    Goal{ Parameter=vdd.dc Voltage= 1.5 }
){ Coupled{ nmos1.poisson  nmos1.electron  nmos1.contact
            pmos1.poisson pmos1.hole      pmos1.contact
            circuit }
}
NewCurrentFile=""
Quasistationary(
    InitialStep=1e-3 Increment=1.35
    MinStep=1e-5 MaxStep=0.05
    Goal{ Parameter=vin.dc Voltage= 1.5 }
){ Coupled{ nmos1.poisson nmos1.electron  nmos1.contact
            pmos1.poisson pmos1.hole      pmos1.contact
            circuit }
}
}

```

在单个器件模拟时，需要解算泊松方程和载流子连续性方程，在混合模拟时，除了解这些方程之外，还需要解其他的方程，例如 `contact` 和 `circuit`。

注：在混合模拟中，默认便会解 `contact` 和 `circuit` 方程。

C.3 瞬态扫描

```

System{
    Vsource_pset vdd (dd 0) { dc = 0.0 }
    Vsource_pset vin (in 0) {
        pulse = (0.0      # dc
                1.5      # amplitude
                10e-12    # td
                10e-12    # tr
                10e-12    # tf
                60e-12    # ton
                140e-12)  # period
    }

    NMOS nmos1 ( "source"=0  "drain"=out "gate"=in "substrate"=0 )
    PMOS pmos1 ( "source"=dd "drain"=out "gate"=in "substrate"=dd )
    Capacitor_pset cout ( out 0 ){ capacitance = 3e-14 }

    Plot "n7_sys_des.plt" (time() v(in) v(out)
                        i(nmos1,out) i(pmos1,out) i(cout,out) )
}

```

```

Solve{
  NewCurrentFile="init"
  Coupled(Iterations=100){ Poisson }
  Coupled{ Poisson Electron Hole Contact Circuit }

  Quasistationary(
    InitialStep=1e-3 Increment=1.35
    MinStep=1e-5 MaxStep=0.05
    Goal{ Parameter=vdd.dc Voltage= 1.5 }
  ){ Coupled{ nmos1.poisson nmos1.electron nmos1.contact
              pmos1.poisson pmos1.hole      pmos1.contact
              circuit }
  }

  NewCurrentFile=""

  Transient (
    InitialTime=0 FinalTime=200e-12
    InitialStep=1e-12 Increment=1.3
    MaxStep=5e-12 MinStep=1e-15
  ){ Coupled{ nmos1.poisson nmos1.electron nmos1.contact
              pmos1.poisson pmos1.hole      pmos1.contact
              circuit }
  }
}

```

C.4 小信号 AC 分析

在小信号分析时，Sdevice 计算 Y 矩阵。Y 矩阵描述当偏压变化较小时，电路中的电流如何变化。

$$\delta \mathbf{i} = \mathbf{Y} \cdot \delta \mathbf{v} = (\mathbf{A} + j \cdot \omega \cdot \mathbf{C}) \cdot \delta \mathbf{v}$$

Y 矩阵可被分为两部分：实部 A 称为 conductance matrix，描述电流随电压的同步变化；虚部 C 称为 Capacitance matrix，描述电流随电压的非同步变化（电流对电压的响应存在延迟）。符号 j 为虚部单位， ω 为小信号变化的频率。

对于典型的四端 MOS 器件（g, d, s, b），A 和 C 具有如下的形式（4x4 的矩阵）：

$$\begin{bmatrix} i(g) \\ i(d) \\ i(s) \\ i(b) \end{bmatrix} = \begin{bmatrix} a(g,g) & a(d,g) & a(s,g) & a(b,g) \\ a(g,d) & a(d,d) & a(s,d) & a(b,d) \\ a(g,s) & a(d,s) & a(s,s) & a(b,s) \\ a(g,b) & a(d,b) & a(s,b) & a(b,b) \end{bmatrix} + j \cdot \omega \cdot \begin{bmatrix} c(g,g) & c(d,g) & c(s,g) & c(b,g) \\ c(g,d) & c(d,d) & c(s,d) & c(b,d) \\ c(g,s) & c(d,s) & c(s,s) & c(b,s) \\ c(g,b) & c(d,b) & c(s,b) & c(b,b) \end{bmatrix} \cdot \begin{bmatrix} v(g) \\ v(d) \\ v(s) \\ v(b) \end{bmatrix}$$

例如， $a(d,g)$ 描述了漏极电流对栅压的小信号变化的同步响应，亦即 MOS 器件的跨导。类似地， $c(b,g)$ 描述了衬底电流对栅压小信号变化的非同步响应，这与栅到衬底的电容有关。

System {

```

NMOS nmos1 (drain=d source=s gate=g substrate=b)
Vsource_pset vd ( d 0 ){ dc = 0 }
Vsource_pset vs ( s 0 ){ dc = 0 }
Vsource_pset vg ( g 0 ){ dc = 0 }
Vsource_pset vb ( b 0 ){ dc = 0 }
}

```

需要进行 AC 分析的每一个电极都需要指定一个电压源。

```

Solve{
  NewCurrentFile="init"
  Coupled(iterations=100){ Poisson }
  Coupled{ Poisson Electron Hole }

  Quasistationary (
    InitialStep=0.1 Increment=1.3
    MaxStep=0.5 Minstep=1.e-5
    Goal { Parameter=vg.dc Voltage=-3}
  ){ Coupled { Poisson Electron Hole } }

  NewCurrentFile=""
  Quasistationary (
    InitialStep=0.01 Increment=1.3
    MaxStep=0.05 Minstep=1.e-5
    Goal { Parameter=vg.dc Voltage=3}
  ){ ACCoupled (
    StartFrequency=1e6 EndFrequency=1e6 NumberOfPoints=1 Decade
    Node(d s g b) Exclude(vd vs vg vb)
    ACCompute (Time = (Range = (0 1) Intervals = 20))
  ){ Poisson Electron Hole }
  }
}

```

Node 语句中的节点列表指定 Y 矩阵中包含哪些节点。与节点列表中的节点相连的所有电压源都必须被排除在 AC 分析之外（采用 **Exclude** 语句），因为对于 AC 信号而言，电压源代表着短路。

ACCompute 语句（可选）强制在预先定义的偏压点上进行 AC 分析。不过，**Sdevice** 也可计算中间点以保证收敛。对于大的结构和收敛慢的模拟，**ACCompute** 可以通过取消中间点的 AC 分析从而加速模拟。

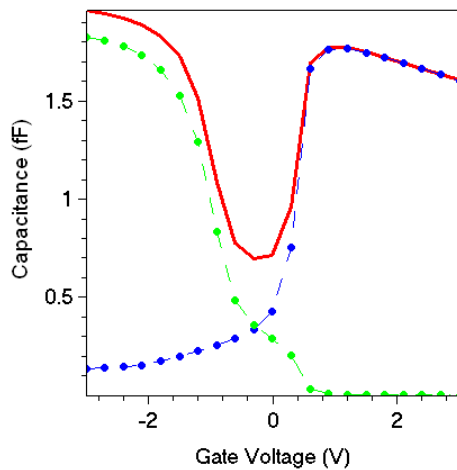


图 4 总电容(red), gate-to-contact (blue)以及 gate-to-body (green) 电容随栅压的变化

D. 一个较复杂的例子（30 分钟）

目标：学习 Sentaurus Device 的高级模拟技术

D.1 MOSFET 的击穿模拟

在击穿之前，漏极电流基本上为常数。此时，采用压控扫描（voltage-controlled bias sweep）能够很好地工作。但是，在击穿之后，一个很小的电压变化都会导致漏极电流的急剧增加，甚至发生 snap-back。此时，采用压控扫描很难收敛，流控扫描便是必需的。

从击穿前的压控扫描转换到击穿后的流控扫描的简单而有效的方法就是在漏极连接一个大小合适的电阻。阻值必须足够大，以保证在击穿后绝大部分电压降落在电阻上，从而电阻/电压源结合在一起便相当于一个电流源。

确定阻值大小的一个简单方法就是在没有外部电阻的情况下进行一个预模拟。通常，这次模拟会在刚击穿或者击穿之后一小段时间失败。将此时的 V_{ds} 除以 I_{ds} ，采用一个量级在 V_{ds}/I_{ds} 附近的电阻。

```
Electrode{
  { Name="source"      Voltage=0.0 }
  { Name="drain"       Voltage=0.0 Resistor= 1e7 }
  { Name="gate"        Voltage=0.0 Barrier= -0.55 }
  { Name="sub"         Voltage=0.0 }
}
```

```
Physics{
  EffectiveIntrinsicDensity( OldSlotboom )
  Mobility(
    DopingDep
    eHighFieldsaturation( GradQuasiFermi )
    hHighFieldsaturation( GradQuasiFermi )
    Enormal
  )
  Recombination(
```

```

SRH( DopingDep )
eAvalanche( Eparallel)
hAvalanche( Eparallel )
)
}

```

eAvalanche 和 **hAvalanche** 用于激活电子和空穴的碰撞电离模型。除了激活碰撞电离模型之外，这两条语句还用于选择计算载流子传输时的驱动力。

在漂移-扩散传输模型中，载流子的驱动力有两种模型：**Eparallel**（默认值）和 **GradQuasiFermi**。在绝大多数情况下，两种模型给出的结果相同或者十分接近。但是，在某些应用中，不同的模型可能导致模拟稳定性的不同。如果发生了收敛性问题，可以尝试选用另一种驱动力模型。

注：在流体力学传输模型中，驱动力为 **CarrierTempDrive**。

```

Math {
  Extrapolate
  Avalderivatives
  Iterations=20
  Notdamped =100
  RelErrControl
  ErRef(Electron)=1.e10
  ErRef(Hole)=1.e10
  BreakCriteria{ Current(Contact="drain" AbsVal=1e-5) }
}

```

Avalderivatives 用于提高击穿模拟的鲁棒性。它将雪崩产生率的导数引入 **Jacobian** 矩阵。如果没有激活碰撞电离模型，请不要使用 **Avalderivatives**，因为这对收敛性没有任何帮助，只会增加内存的使用量和运行时间。

可以指定电流或者电压在达到某个值之后结束模拟。本例中设置漏极电流超过 **1e-5A/um** 后停止模拟。

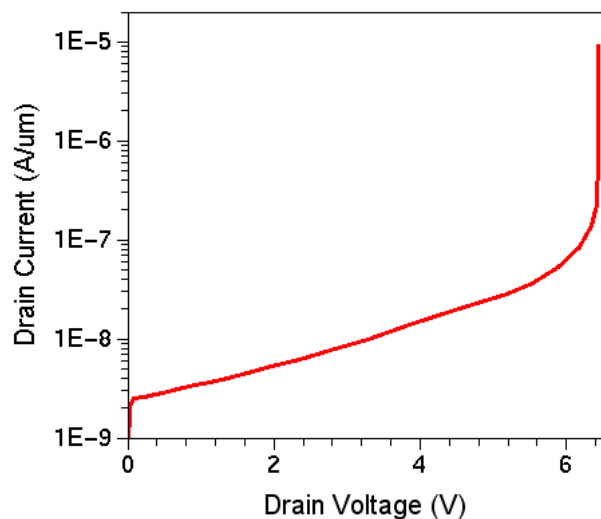


图 1 NMOS 的关态击穿($V_{gs}=0$)

D.2 电流边界条件

本例阐述流控模拟时的典型输入命令。

本例模拟三极管的 **IcVc** 特性。在模拟时，首先将基极电流 **ramp** 到某几个预定义的值，结果保存在不同

的文件中。在模拟的第二个阶段，将在前一步保存的文件，并进行集电极的电压扫描。

```
Physics{
    Recombination(
        SRH(DopingDep)
        Auger
    )
    Mobility( DopingDep HighFieldSaturation)
    EffectiveIntrinsicDensity( OldSlotboom )
}
```

注：

- 此处激活了俄歇复合模型。
- 由于表面粗糙度散射导致的迁移率下降（Mobility 语句中的 Enormal）在绝大部分 MOSFET 模拟时是一个标准做法，但是对于三极管是不需要考虑的。

```
Math{
    Extrapolate
    RelErrControl
    Digits=5
    ErrRef(electron)= 1.0e7
    ErrRef(hole)      = 1.0e7
    Iterations=20
    Notdamped=100
}
```

ErrRef 语句定义了误差控制的参考值。本例中的值 1.0e7 表示载流子浓度超过 1.0e7/cm³ 的节点对误差估计的贡献是必须考虑的，而载流子浓度低于 1.0e7/cm³ 的节点则不怎么考虑。默认值为 1.0e10。如果载流子密度较低的区域对器件的工作影响很大，可适当将默认值 1.0e10 减小。

```
Solve {
    NewCurrentFile="init"
    Coupled(Iterations=100){ Poisson }
    Coupled{ Poisson Electron Hole }

    Quasistationary (
        InitialStep=0.01 Increment=1.3
        MaxStep =0.2 MinStep = 1e-6
        Goal{ Name="base" Voltage=0.5 }
    ){ Coupled {Poisson Electron Hole } }
```

在获得初始解之后，基极电压 ramp 到 0.5V。

```
Set( "base" mode current )
```

```
Quasistationary (
    InitialStep=0.01 Increment=1.3
    MaxStep =0.2 MinStep = 1e-6
    Goal{ Name="base" Current=1e-8 }
){ Coupled {Poisson Electron Hole } }
```

Set 命令将基极的边界条件从电压控制改变为电流控制。

注：在由电压控制改变为电流控制之前，基极电压首先 ramp 到 0.5V，远离平衡条件。这是因为，在平

衡时，对于关闭的 pn 结而言，很小的电流改变都会导致很大的电压跳变，这可能导致收敛变慢。另外，由于可能的小电流波动，零电流边界条件有时本身就导致收敛变慢。通常，一个好的注意是对器件施加偏置，使其电流不为零，从而避免这类收敛性问题。

Quasistationary 命令用于将基极电流 ramp 到 $1\text{e-}8\text{ A}/\mu\text{m}$ 。

```
Quasistationary (
  InitialStep=0.01 Increment=1.3
  MaxStep =0.2 MinStep = 1e-6
  Goal{ Name="base" Current=1e-7 }
){ Coupled {Poisson Electron Hole }
  Save(FilePrefix="n2_ib" Time=(0.0;0.5;1) NoOverWrite )
}
```

```
NewCurrentFile="Ib0"
Load(FilePrefix="n2_ib_0000")
```

```
Quasistationary (
  InitialStep=0.01 Increment=1.3
  MaxStep =0.2 MinStep = 1e-6
  Goal{ Name="collector" Voltage=1.0 }
){ Coupled {Poisson Electron Hole } }
```

```
NewCurrentFile="Ib1"
Load(FilePrefix="n2_ib_0001")
```

```
Quasistationary (
  InitialStep=0.01 Increment=1.3
  MaxStep =0.2 MinStep = 1e-6
  Goal{ Name="collector" Voltage=1.0 }
){ Coupled {Poisson Electron Hole } }
```

```
NewCurrentFile="Ib2"
Load(FilePrefix="n2_ib_0002")
```

```
Quasistationary (
  InitialStep=0.01 Increment=1.3
  MaxStep =0.2 MinStep = 1e-6
  Goal{ Name="collector" Voltage=1.0 }
){ Coupled {Poisson Electron Hole } }
```

```
}
```

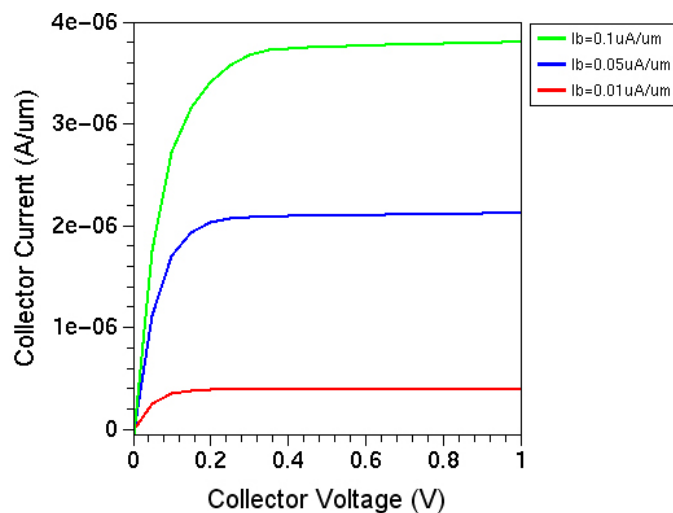


图 2 不同基极电流下，集电极电流随集电极电压的变化。

G. 在 SWB 中运行 SDevice（25 分钟）

G.1 简介

本节重点突出 Sdevice 的独立输入文件在 SWB 中运行时需要进行哪些修改（为了使用 SWB 的预处理功能）。

本节突出了输入命令文件中与 SWB 预处理相关的部分。

G.2 File Section

特定的文件名采用预定义的 SWB 文件名占位符代替。

```
File{
#-Predefined SWB parameters
    Grid    = "@tdr@"
    Plot    = "@tdrdat@"
    Current = "@plot@"
    Output  = "@log@"
    Param   = "@parameter@"
}
```

在预处理时，SWB 将占位符采用实际的文件名替换，具体规则如下：

对于输入文件 @tdr@，SWB 搜索树中的前一节点（一般是 SDE 或者 Mesh，产生器件结构）。SWB 将占位符用前一节点的器件结构代替，如 n2_msh.tdr。

注：如果是 DF-ISE 格式，则采用

```
Grid = "@grid@"
Doping = "@doping@"
```

注：如果 SDevice 是工具流中的第一个工具，输入文件必须显式指定。输出文件的格式为 n3_des.tdr, n3_des.plt, n3_des.log。总之，输出文件有一个固定的命名规范，以方便工具流中的下一个工具使用。

注：如果是 DF-ISE 格式，使用 Plot = "@dat@"。

对于@parameter@文件，SWB 搜索名为 sdevice_Si.par 的文件。SWB 对该文件进行预处理，例如变为 pp3_des.par，然后将@parameter@占位符替换为 pp3_des.par。

注：在 stand-alone 模拟时，参数文件可以采用任何名称。但是，在 SWB 中是，参数文件的名称必须为 sdevice_Si.par，否则参数文件不会被预处理。（Q：参数文件被预处理后，内容是否会发生变化？亦即 sdevice_Si.par 和 pp3_des.par 二者内容是否完全一致？）

G.3 使用 SWB 参数

```
Solve {
  NewCurrentFile="init"
  Coupled(Iterations=100){ Poisson }
  Coupled{ Poisson Electron Hole }

  Quasistationary(
    InitialStep=1e-1 Increment=1.2
    Minstep=1e-5 MaxStep=0.1
    Goal{ Name="drain" Voltage= @Vd@ }
  ){ Coupled{ _EQUATIONS_ } }

  NewCurrentFile=""

  Quasistationary(
    DoZero
    InitialStep=5e-2 Increment=1.5
    Minstep=1e-5 MaxStep=0.05
    Goal{ Name="gate" Voltage= @Vdd@ }
  ){ Coupled{ _EQUATIONS_ } }
#-Arithmetic with variables: enforcing ~0.1V steps for all values of Vdd
  CurrentPlot( Time=(Range=(0 1) Intervals=@<int(Vdd/0.1)>@ )
  )
}
}

字符串参数:
Electrode{
  { Name="source"      Voltage= 0.0 Resistor= 40 }
  { Name="drain"       Voltage= 0.0 Resistor= 40 }
  { Name="gate"        Voltage=-0.1 }
  { Name="substrate"   Voltage= 0.0 }
#-Conditional inclusion of code:
  #if [string compare @Body@ "Tied"] == 0
    { Name="bodytie"    Voltage=0.0 eRecVelocity=0 }
  #endif
}
```

G.4 算术表达式

```
Solve {
    NewCurrentFile="init"
    Coupled(Iterations=100){ Poisson }
    Coupled{ Poisson Electron Hole }

    Quasistationary(
        InitialStep=1e-1 Increment=1.2
        Minstep=1e-5 MaxStep=0.1
        Goal{ Name="drain" Voltage= @Vd@ }
    ){ Coupled{ _EQUATIONS_ } }

    NewCurrentFile=""

    Quasistationary(
        DoZero
        InitialStep=5e-2 Increment=1.5
        Minstep=1e-5 MaxStep=0.05
        Goal{ Name="gate" Voltage= @Vdd@ }
    ){ Coupled{ _EQUATIONS_ } }
#-Arithmetic with variables: enforcing ~0.1V steps for all values of Vdd
    CurrentPlot( Time=(Range=(0 1) Intervals=@<int(Vdd/0.1)>@ )
    )
}
```

算术表达式在预处理时被求值。

G.6 参数文件的参数化

```
AvalancheFactors
{ * Coefficientss for avalanche generation with hydro
  * Factors n_l_f, p_l_f for energy relaxation length in the expressions
  * for effective electric field for avalanche generation
  * eEeff = eEeff / n_l_f ( or b = b*n_l_f )
  * hEeff = hEeff / p_l_f ( or b = b*p_l_f )
  * Additional coefficients n_gamma, p_gamma, n_delta, p_delta
  * n_l_f = 1 # [1]
  * p_l_f = 1 # [1]
  * n_l_f = @AvalFac@ # [1]
  * p_l_f = @AvalFac@ # [1]
  * n_gamma = 1 # [1]
```

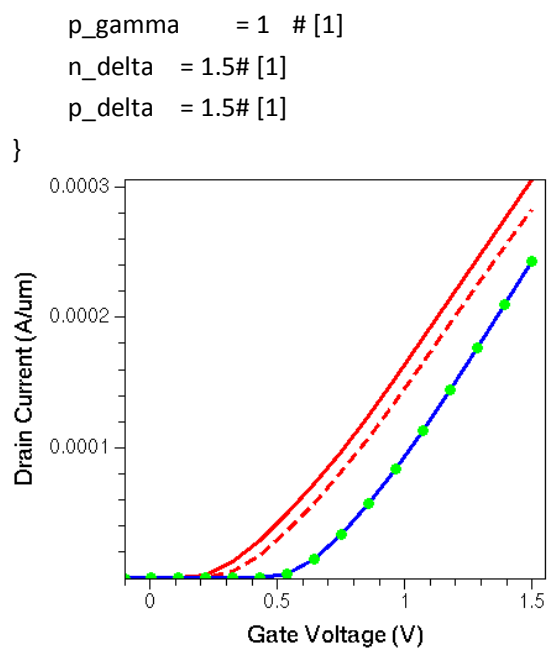


图 1 漏极电流随漏极电压的变化。红色曲线为浮体 SOI NMOSFET，蓝色线和绿色点为带体引出的器件。将 *AvalFac* 的值从 1（实线）变为 1.5（虚线和绿色点）时，对带体引出的器件没有影响。但是，对于浮体器件而言，碰撞电离率的改变将影响体电势和电流。