

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ»

**ОСНОВЫ РАБОТЫ В СРЕДЕ
ПРИБОРНО-ТЕХНОЛОГИЧЕСКОЙ САПР
SENTAURUS**

Учебно-методическое пособие

Воронеж
Издательский дом ВГУ
2017

Утверждено научно-методическим советом физического факультета
29 июня 2017 года, протокол № 6

Составители: Р.П. Алексеев, Е.Н. Бормонтов, Г.В. Быкадорова, А.Ю. Ткачѳв, А.Н. Цоцорин

Рецензент – доктор физико-математических наук, профессор кафедры физики твердого тела и наноструктур С.И. Курганский

Учебно-методическое пособие подготовлено на кафедре физики полупроводников и микроэлектроники физического факультета Воронежского государственного университета.

Рекомендовано для студентов физического факультета очной формы обучения по программам бакалавриата и магистратуры.

Для направлений: 03.03.03, 03.04.03 – Радиофизика; 11.03.04, 11.04.04 – Электроника и наноэлектроника

СОДЕРЖАНИЕ

1. Состав пакета САПР SENTAURUS и назначение компонентов	5
2. Программа-оболочка Workbench	9
3. Модуль технологического моделирования SENTAURUS Process	19
3.1. Назначение модуля SProcess	19
3.2. Структура командного файла модуля SProcess	20
3.3. Основные команды модуля SProcess и их параметры	20
3.3.1. Подраздел определения переменных	21
3.3.2. Подраздел настройки используемых моделей	21
3.3.3. Подраздел настройки сетки	25
3.3.4. Подраздел определения фоторезистивных масок	28
3.3.5. Подраздел последовательности технологических операций .	29
3.3.6. Подраздел преобразования и сохранения структуры	35
3.4. Пример командного файла модуля SProcess	37
4. Модуль структурного проектирования SENTAURUS Structure Editor	44
4.1. Назначение модуля SDE	44
4.2. Структура командного файла модуля SDE	44
4.3. Основные команды модуля SDE и их параметры	45
4.3.1. Определение переменных	46
4.3.2. Определение геометрических размеров и координат Областей	47
4.3.3. Задание профилей распределения примесей	48
4.3.4. Определение контактов	49
4.3.5. Построение расчетной сетки	50
4.4. Пример командного файла модуля SDE	51
5. Оптимизатор расчетной сетки SNMesh	57
5.1. Назначение программного модуля SNMesh	57
5.2. Запуск и использование модуля SNMesh	57
5.3. Структура командного файла модуля SNMesh	60
5.4. Пример командного файла модуля SNMesh	61

6. Программный модуль SENTAURUS Device для моделирования параметров полупроводниковых приборов	64
6.1. Назначение модуля SDevice	64
6.2. Структура командного файла модуля SDevice	66
6.3. Основные команды модуля SDevice	67
6.4. Пример командного файла модуля SDevice	76
7. Программа Inspect для просмотра и обработки графиков	78
8. Универсальная программа Techplot SV для просмотра результатов моделирования	84
9. Универсальная программа отображения информации SENTAURUS Visual	89
Библиографический список	95

1. СОСТАВ ПАКЕТА САПР SENTAURUS И НАЗНАЧЕНИЕ КОМПОНЕНТОВ

САПР **SENTAURUS** фирмы Synopsys (США) относится к числу известных и широко используемых во всем мире систем моделирования. Многие крупные фирмы, занимающиеся производством полупроводниковых приборов, пользуются пакетом **SENTAURUS**, который позволяет не только проектировать полупроводниковые приборы, но и разрабатывать новые технологии, исследовать физические процессы в полупроводниках. Фактически эта система моделирования является пакетом как для научных исследований, так и виртуального производства.

На сегодняшний день разработано множество программных продуктов для моделирования процессов микроэлектроники, таких, например, как **SILVACO** (США), **MICROTEC** (Канада).

В 2004 году компания Synopsys, один из мировых лидеров в области САПР СБИС, объединилась с швейцарской компанией ISE AG, занимавшей ведущие позиции в сфере приборно-технологического моделирования. В октябре 2005 года появилась система **SENTAURUS TCAD**, объединившая в себе преимущества средств приборно-технологического проектирования обеих компаний.

SENTAURUS TCAD сегодня – это динамично развивающийся продукт с периодичностью обновления девять месяцев. Основная задача системы – обеспечение разработчиков средствами приборно-технологического проектирования нового поколения, а также интеграция этих средств с САПР СБИС.

Развитие средств приборно-технологического проектирования сегодня сосредоточено на следующих направлениях:

- трехмерное моделирование субмикронных приборов, включающее моделирование технологического процесса формирования структуры прибора, механических напряжений внутри прибора и анализ трехмерного распределения носителей заряда;

- моделирование мощных кремниевых и гетероприборов, в том числе на основе SiC и GaN, приборов на основе материалов A^3B^5 , использующих гетеропереходы, фотодетекторов, светоизлучающих диодов и полупроводниковых лазеров;

- построение компактных моделей на базе результатов моделирования или измерений конкретного прибора, отражающих зависимость выходных

электрофизических параметров от разброса входных технологических параметров.

В современные версии добавлены новые модели технологических процессов, такие, как модели лазерного и импульсного отжига примесей, улучшенные модели диффузии примесей и др. .

В новых версиях TCAD реализована полная поддержка моделирования технологии и электрических характеристик приборов на широкозонных полупроводниках – SiC, GaN и др. , а также существенно переработан математический блок решения систем дифференциальных уравнений. Это позволило значительно улучшить сходимость уравнений при решении таких сложных в вычислительном плане задач, как, например, расчет выходной вольт-амперной характеристики транзисторов и диодов, а также моделирование электрофизических характеристик приборов на основе широкозонных полупроводников.

В настоящее время в **SENTAURUS** TCAD реализовано распараллеливание и многопоточность расчета, что позволяет полностью использовать возможности современных многоядерных процессоров и существенно сокращать затраты машинного времени на моделирование. Разработан новый генератор высококачественной расчетной сетки, адаптирующейся под конструктивные особенности прибора и профили распределения примесей. Использование этого генератора взамен устаревших позволяет существенно увеличить точность и скорость расчета, а также улучшить сходимость систем уравнений.

В последние версии включены средства разработки технологических SPICE-моделей, а также программы для анализа влияния разброса технологических параметров на электрические характеристики. Добавлены средства моделирования, позволяющие определять оптимальные технологические параметры для снижения чувствительности электрических характеристик приборов к технологическим разбросам.

В состав приборно-технологической САПР **SENTAURUS** входят более 30 программных модулей для моделирования технологии и топологии, структур, сеток и электрофизических параметров полупроводниковых приборов (рис. 1.1).



Рис. 1.1. Состав приборно-технологической САПР SENTAURUS

Организацию процесса моделирования обеспечивает графическая управляющая оболочка **SENTAURUS Workbench (SWB)**, интегрирующая программные модули, внутренний интерфейс между ними и конечным пользователем.

Основной программный модуль **SENTAURUS Process (SProcess)** предназначен для одно-, двух- и трёхмерного сквозного моделирования технологических маршрутов при проектировании и изготовлении полупроводниковых структур различного функционального назначения. Наряду

с **SProcess** в **SENTAURUS** есть еще три дополнительных модуля приборно-технологического моделирования: **Dios**, **Suprem-IV** и **Taurus Process**.

Генерацию высококачественных одно-, двух- и трёхмерных конечно-элементных сеток для последующего моделирования приборов обеспечивает следующий набор модулей: модуль **Mesh** генерирует одно-, двух- и трёхмерную сетку, параллельную координатным осям; модуль **Noffset3D** создаёт триангулярную сетку для двумерных задач, и тетраэдральную – для трёхмерных; новый устойчивый генератор сеток **SENTAURUS Mesh (SNMesh)** позволяет создавать сетки, параллельные осям, а также сетки тензорного типа.

Модуль **SENTAURUS Structure Editor (SDE)** для графического проектирования (конструирования) двумерных и трехмерных полупроводниковых структур исключает применение программ технологического моделирования. Формирование структуры при этом включает в себя генерацию геометрической модели по слоям структуры вместе с контактами, задание аппроксимирующих профилей легирования и определение процесса построения вычислительной конечно-элементной сетки.

Основной модуль **SENTAURUS Device (SDevice)** предназначен для двумерного и трехмерного моделирования полупроводниковых приборов с учетом различных электрофизических приближений: диффузионно-дрейфового, гидродинамического, с учетом квантовых поправок, механических напряжений и т. д. Обеспечивает расчет, анализ и оптимизацию различного рода параметров и характеристик (электрофизических, тепловых, оптических и т. д.) для широкого ряда полупроводниковых структур от кремниевых MOS-нанотранзисторов и мощных биполярных транзисторов до гомо- и гетероструктур на основе сложных материалах типа A^3B^5 , карбиде кремния и т. д. Кроме модуля **SDevice** в блок модулей электрофизического моделирования приборов **SENTAURUS** входят также: **SMoca** и **Sparta** – для моделирования субмикронных приборов методом Монте-Карло; **SDevice Electromagnetic Wave** – для моделирования электромагнитных процессов электродинамики.

Программные модули **Tecplot SV**, **Inspect**, **Measure**, **SVisual** предназначены для визуального отображения результатов моделирования, расчета параметров и характеристик моделируемых полупроводниковых приборов.

Этих программных модулей и программ достаточно для решения многих задач, связанных с исследованием и разработкой полупроводниковых приборов.

2. ПРОГРАММА-ОБОЛОЧКА WORKBENCH

Проект в САПР **SENTAURUS** создается в программе-оболочке **Workbench (SWB)**, запуск которой осуществляется щелчком мыши на соответствующем значке панели задач рабочего стола, либо командной строкой **swb**.

Главное окно программы **Workbench** (рис. 2.1) содержит строку меню, строку кнопок управления, менеджер проектов, рабочую область с двумя вкладками отображение структуры проекта **Project** и управления работающими программами **Scheduler**, а также строку состояния, на которой показаны условные цветовые обозначения статусу узлов.

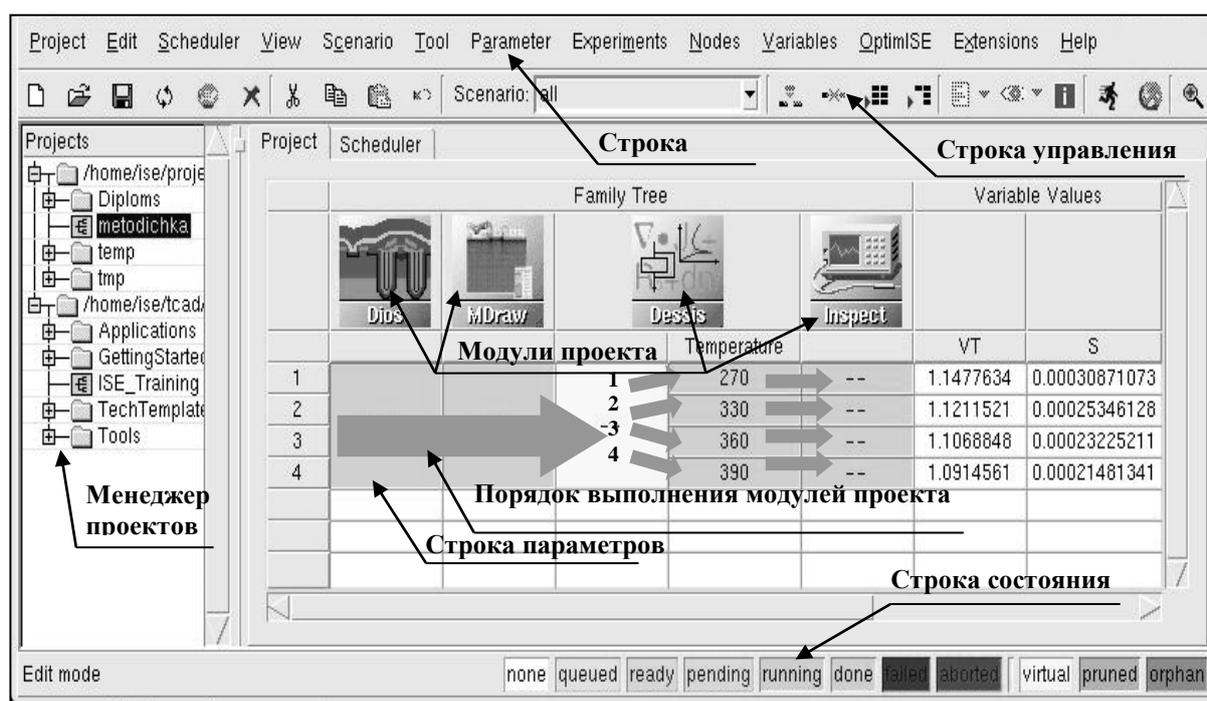


Рис. 2.1. Главное окно **SENTAURUS Workbench**

Проект представляет собой последовательность модулей и программ САПР **SENTAURUS**, управляемых командными файлами, и имеет структуру дерева, в простейшем случае представляющего собой всего одну ветвь. Программы в проекте выполняются последовательно слева направо вдоль каждой ветви, используя результаты моделирования предыдущих программ.

Строки меню и кнопок управления имеют стандартный вид и общепринятое назначение компонентов. Наиболее часто из них используются следующие элементы:

- пункты строки меню:

- **Project** – работа с проектом в целом: позволяет открыть, создать, сохранить проект и др.;
- **View** – настройка вида окна;
- **Tool** – работа с модулями: добавление, удаление, блокировка и др.;
- **Experiments** – работа с экспериментами: добавление, удаление и др.;
- **Nodes** – работа с узлами: выделение, блокировка, запуск и др.;
- **Variables** – работа с параметрами: добавление, формат и др.;
- **Extensions** – позволяет запускать модули и программы **SENTAURUS**;
- **Help -> Manuals** – вызов Руководства пользователя с подробным описанием всех модулей и поддерживаемых ими команд;

- кнопки строки управления:

-  – создание нового проекта;
-  – открытие сохраненного ранее проекта;
-  – сохранение текущего проекта;
-  – обновление рабочей области;
-  – закрыть проект;
-  – вырезать выбранный эксперимент;
-  – копировать выбранный проект;
-  – отмена предыдущего действия;
-  – вставить новый эксперимент (по сути эквивалентно добавлению);
-  – удалить выбранный эксперимент;
-  – добавить новый эксперимент;
-  – добавить диапазон значений параметров (серию экспериментов);
-  – визуализация результатов расчета выбранного узла;
-  – запуск расчета выделенного проекта или узла/узлов;
-  – прерывание расчета текущего проекта или узла/узлов;
-  – меняет способ отображения дерева – эксперименты располагаются по столбцам, а программные модули – строкам;
-  – вызов командной строки;
-  – вызов Руководства пользователя;

Менеджер проектов представляет собой файловый менеджер, который позволяет выполнять такие стандартные действия, как копирование, перемещение, удаление проектов и директорий и т.п. Проекты отображаются в виде дерева в соответствии с их расположением в файловой системе. Открыть проект можно двойным щелчком левой кнопки мыши. Щелчок правой кнопки мыши на проекте вызывает контекстное меню (рис. 2.2), с помощью которого можно запустить *run* либо прервать *abort* расчет проекта, разблокировать его *unlock*, удалить результаты расчетов, т.е. очистить проект *clean up*, просмотреть протокол выполнения *view log* и др.

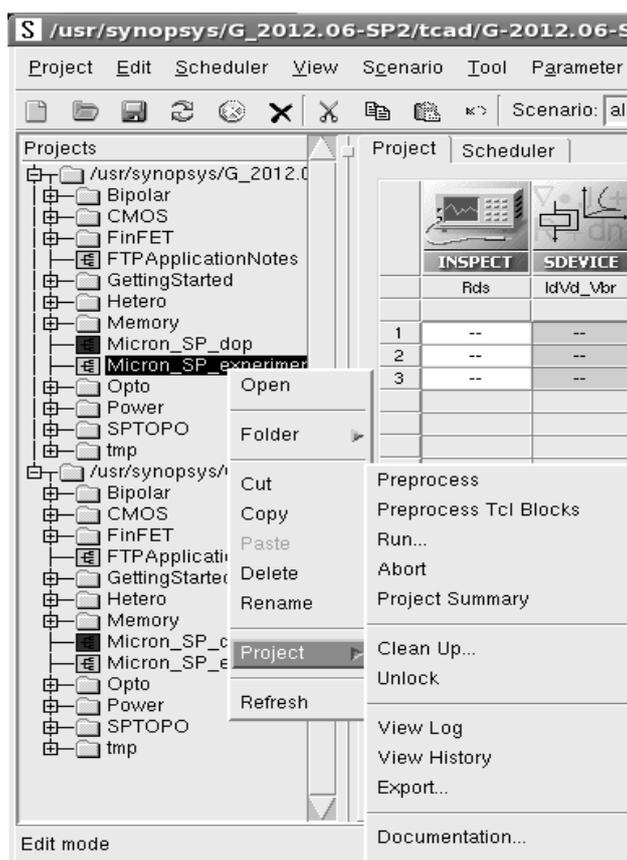


Рис. 2.2. Контекстное меню проекта

При выборе пункта *Clean up* появляется окно (рис. 2.3), позволяющее выбрать одно или несколько вариантов очистки. Особое внимание стоит обратить на следующие из них:

- *Renumber the Tree* – перенумерация узлов дерева. Рекомендуется в случае, если было произведено удаление/добавление части ветвей или модулей в начале или середине дерева;
- *Output Files* – удаление всех расчетов;

- **Extracted Variables** – удаление рассчитанных параметров;
- **Preprocessed Files** – удаление файлов, созданных **SWB** при компиляции проекта, в том числе копии командного файла для каждой ветки.

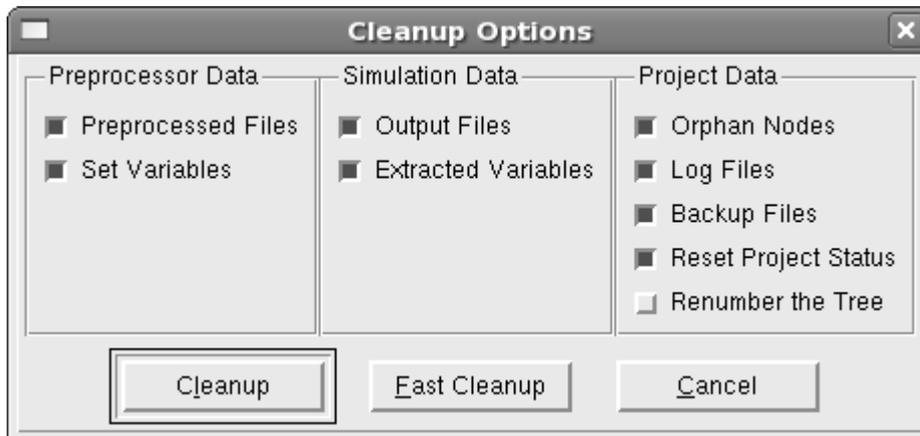


Рис. 2.3. Меню *Clean Up*

Рабочая область **SWB** используется для создания, редактирования и запуска расчетов проектов, а также для управления запущенными проектами. В начале создания нового проекта рабочая область пуста и имеет вид, показанный на рисунке 2.4.

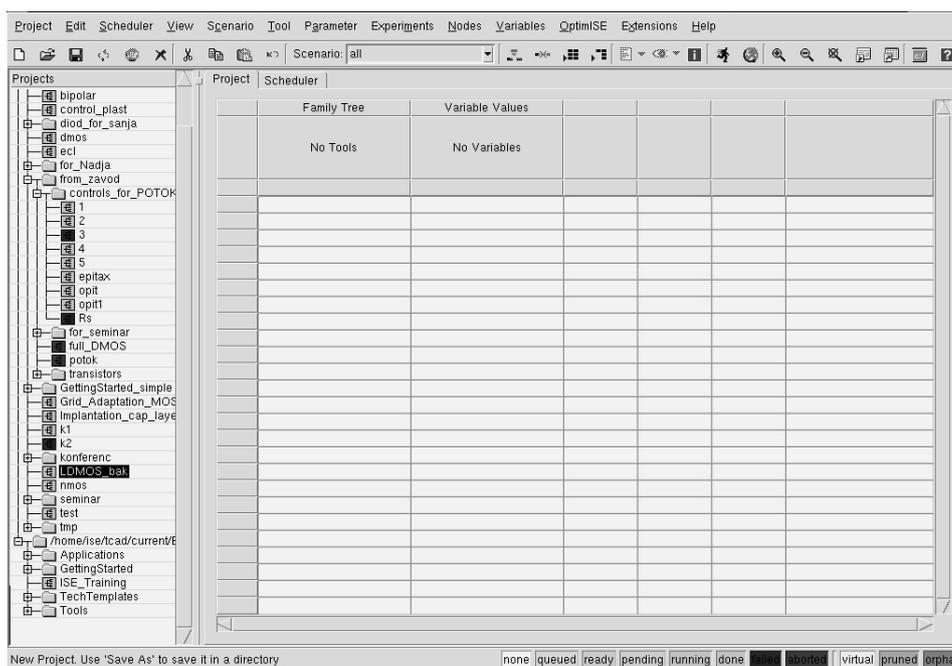


Рис. 2.4. Пустой проект в главном окне *SENTAURUS Workbench*

Для добавления программных модулей в создаваемый проект нужно правой кнопкой мыши щелкнуть на области пиктограмм программных модулей рабочей области **No Tools** и в появившемся контекстном меню выбрать пункт **Add**. В открывшемся диалоговом окне (рис. 2.5а) вписать название программного модуля, либо нажать на кнопку **Tools** и выбрать этот пункт из пиктограмм (рис. 2.5б). В запросе об используемом сценарии выбирается параметр **default**. Далее для добавления программных модулей к проекту нужно щелкать правой кнопкой мыши на пиктограмме одной из уже имеющихся программ и повторять вышеописанные действия. При этом следует обратить внимание на переключатель **after selected tools/ before selected tools**, который появляется в диалоге выбора программного модуля. Новый модуль будет добавлен соответственно после или до того программного модуля, по пиктограмме которого щелкали мышью.

Для редактирования командного файла какого-либо программного модуля нужно щелкнуть правой кнопкой мыши на соответствующей иконке проекта в программе-оболочке **SWB** и в появившемся меню (рис. 2.6) выбрать **Edit Input -> Commands**. В результате запустится текстовый редактор **ISEEdit**, с помощью которого можно набирать и редактировать командные файлы.

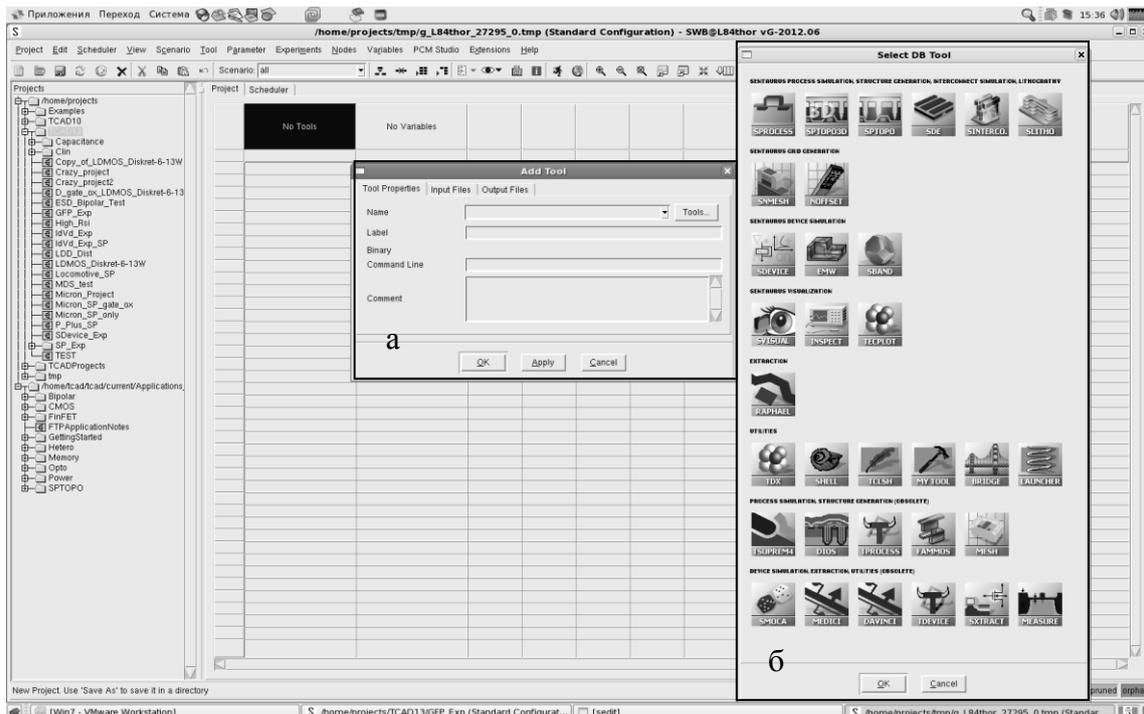


Рис. 2.5. Добавление модулей и программных блоков к новому проекту:
 а – диалоговое окно контекстного меню **Add**;
 б - набор программных модулей **TCAD SENTAURUS**

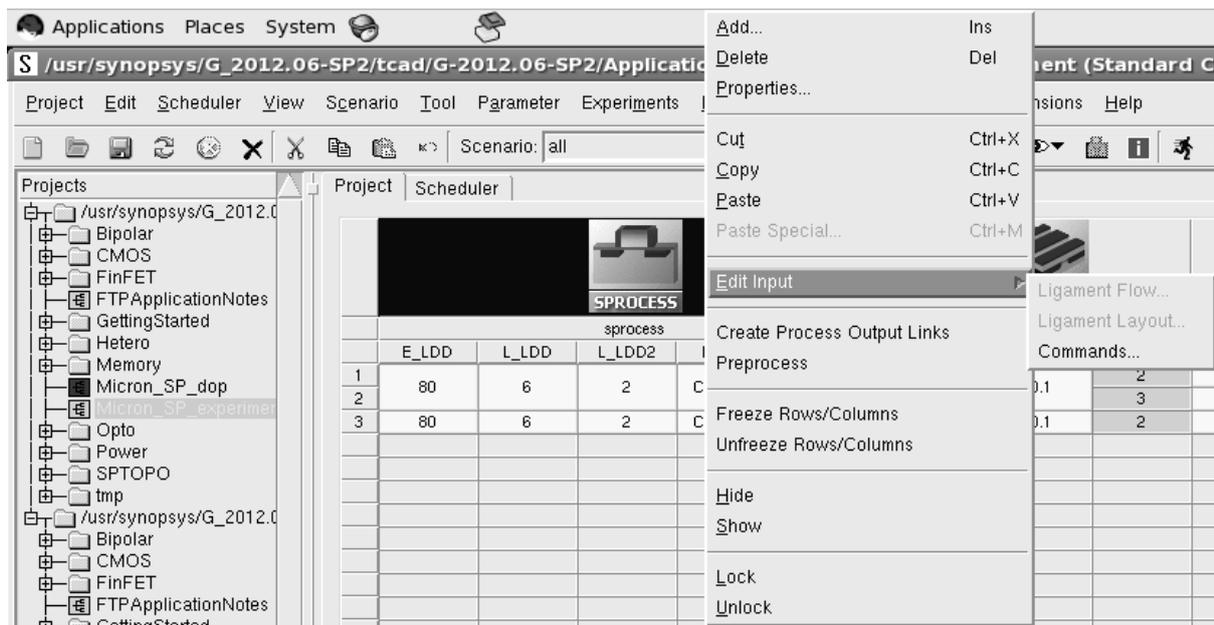


Рис. 2.6. Вызов редактора командных файлов

Для добавления параметров к какому-либо программному модулю необходимо щелкнуть правой кнопкой мыши на ячейке строки параметров под соответствующим программным модулем. В вызванном меню нужно выбрать **Add**, затем в диалоговом окне указать имя параметра и его значение по умолчанию (рис. 2.7).

Для добавления значений параметра нужно выполнить аналогичные действия, но выбрать пункт **Add Values**. Далее нужно указать, начиная с какого значения, с каким шагом и сколько добавлять значений с учетом начального (рис. 2.8).

Также можно добавлять целые наборы значений различных параметров (эксперименты). Для этого стоит выбрать одну из строк узлов, вызвать контекстное меню и выбрать **Add New Experiments**, либо нажать клавишу **Insert**, либо на иконку . В появившемся окне можно изменить любое количество параметров (рис. 2.9).

Параметры используются для того, чтобы передать какие-либо данные в соответствующий программный модуль. Например, если необходимо смоделировать ионное легирование в **SProcess** меняя энергию ионов в заданных пределах с некоторым шагом, то это гораздо удобнее сделать, используя параметр в программе-оболочке **SWB** с соответствующим набором значений, чем делать несколько проектов, отличающихся лишь энергией имплантируемых ионов. При использовании параметров, задание

которых в командных файлах выглядит как @имя_параметра@, программа-оболочка **SWB** во время препроцессирования проекта создает необходимое количество командных файлов, в которых вместо параметра подставлено соответствующее значение, и программный модуль будет запускаться столько раз, сколько значений у параметра, т.е. с каждым получившимся после препроцессирования командным файлом. Это и есть ветвление проекта (рис. 2.10). Внешне запуск расчета каждого командного файла выглядит как запуск узла проекта с нужным значением параметра.

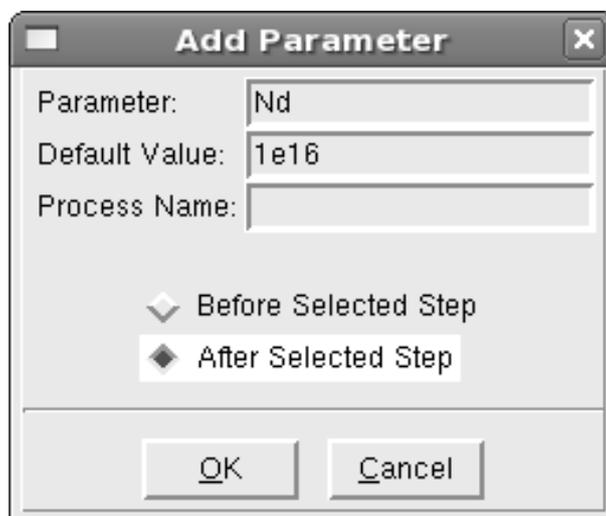


Рис. 2.7. Диалог добавления параметра

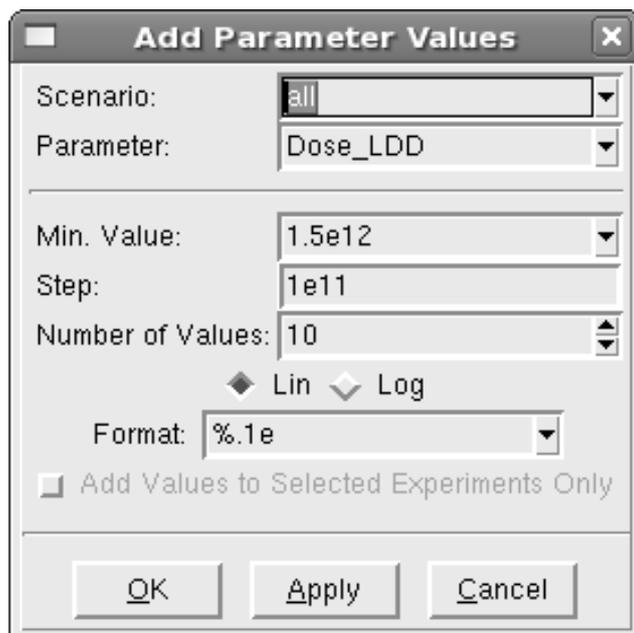


Рис. 2.8. Диалог добавления значений параметра

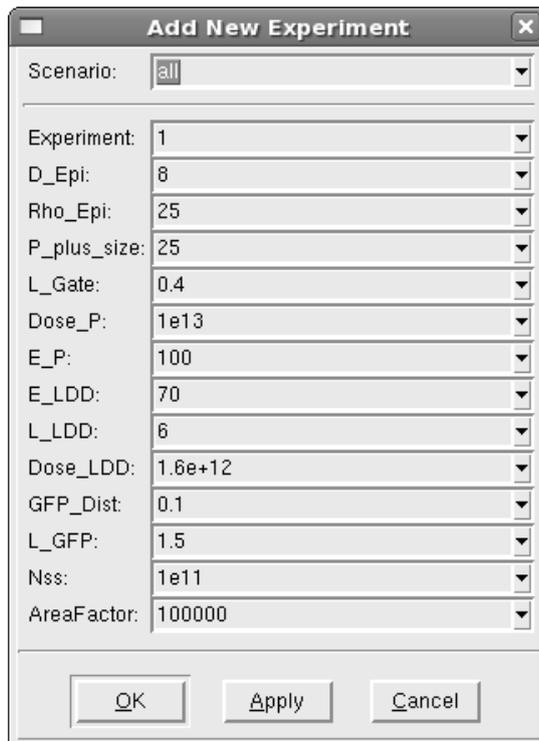


Рис. 2.9. Диалог добавления нового эксперимента

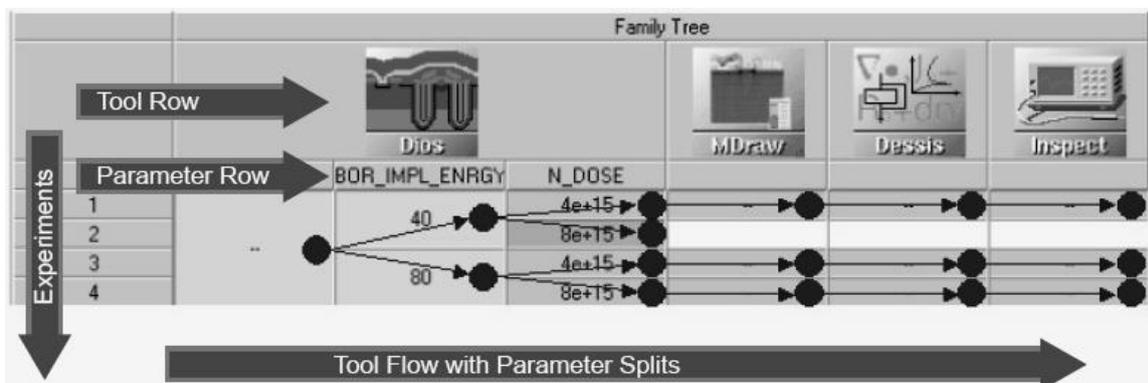


Рис. 2.10. Порядок выполнения узлов проекта.
Ветвление проекта по параметрам *BOR_IMPL_ENRGY* и *N_DOSE*

Значение параметра какого-либо конкретного узла можно изменить, выбрав данный узел и нажав клавишу F6.

Запуск расчета узлов осуществляется в следующей последовательности: выделить левой кнопкой мыши нужный узел или несколько узлов, используя клавиши **Control** или **Shift** клавиатуры, щелкнуть мышью на кнопке  строки кнопок управления либо щелкнуть правой кнопкой на выделенных узлах и выбрать в меню **Run**. В появившемся диалоговом окне (рис.2.11) отображаются номера узлов, которые будут запущены.

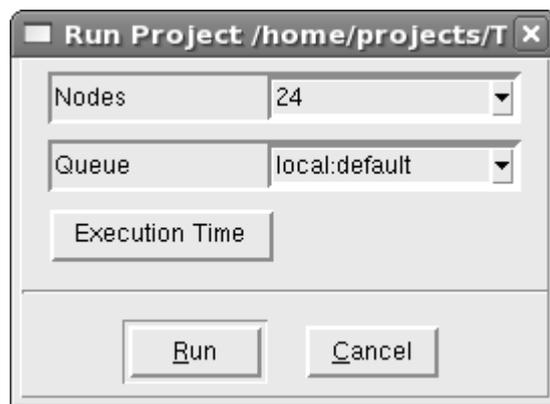


Рис. 2.11. Диалоговое окно запуска расчета узлов проекта

По умолчанию расчет будет вестись на данном локальном компьютере (*Queue: local:default*), хотя при наличии кластера расчет можно провести на нем, что значительно быстрее. Если ни один из узлов не выделен, то будет запущен расчет всего проекта, порядок расчета узлов которого показан на рисунке 2.10. Порядковые номера узлов можно просмотреть, нажав **Ctrl+2**. Вернуться можно, нажав **Ctrl+1**. По умолчанию будут рассчитываться узлы, которые не были рассчитаны *remaining*, хотя можно пересчитать весь проект целиком *all*. Для прерывания расчета используется кнопка . Статус узла – рассчитан, прерван, ошибка, не рассчитан и т.д. – обозначается цветом.

Если щелкнуть правой кнопкой по какому-либо узлу, появится контекстное меню (рис. 2.12), наиболее часто используемые пункты которого: **Run** – запустить расчет; **Abort** – прервать выполнение; **View Output** – просмотреть сообщения программы о ходе расчета); **Visualize** – просмотр результатов экспериментов. В подменю **Visualize** (рис. 2.13) можно выбрать программу визуализации: **Tecplot SV** – для визуального представления результатов расчетов; **Inspect** – для просмотра данных в виде графиков; **SEN-TAURUS Visual** – способен выполнить обе эти задачи, и др. Остальные пункты меню используются редко. Меню **Visualize** также может быть вызвано нажатием на кнопку .

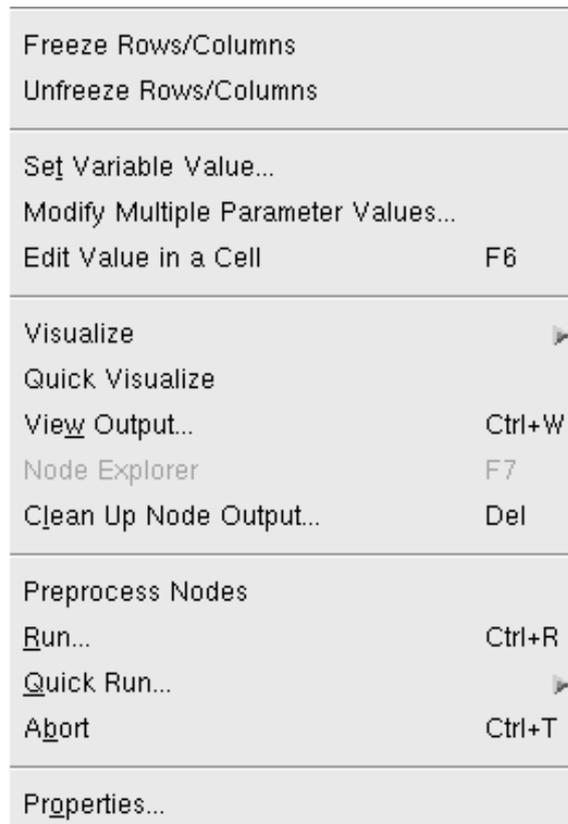


Рис. 2.12. Контекстное меню для выделенного узла

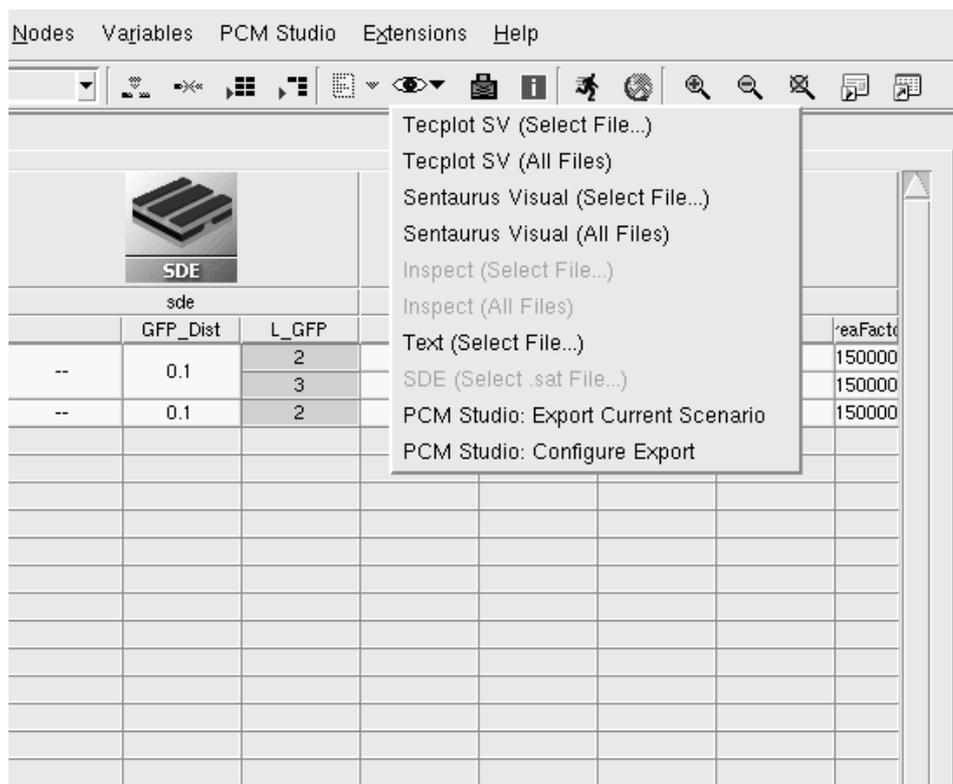


Рис. 2.13. Подменю **Visualize** для выделенного узла

3. МОДУЛЬ ТЕХНОЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ SENTAURUS PROCESS

3.1. Назначение модуля SProcess

Программа **Sentaurus Process (SProcess)** предназначена для моделирования технологии изготовления полупроводниковых приборов. В качестве входных данных используется либо ранее смоделированная структура, либо новая структура с параметрами, указанными в командном файле. Управление ходом моделирования осуществляется с помощью командного файла, в котором описана вся технология изготовления прибора, топологические параметры, применяемые математические модели и другие данные. Результатом моделирования является виртуальная структура, представляющая собой сетку, в узлах которой указаны значения таких параметров, как тип материала, тип примесей, их концентрации и т.д. Эти данные заносятся в один или несколько файлов.

Как и все программы пакета **SENTAURUS TCAD**, **SProcess** удобно использовать, создав проект в программе-оболочке **Workbench**. Вызвав диалоговое окно и выбрав в нем вкладку **Tool Properties for sprocess** (рис. 3.1), можно установить параметры запуска **SProcess**: графический или фоновый режим.

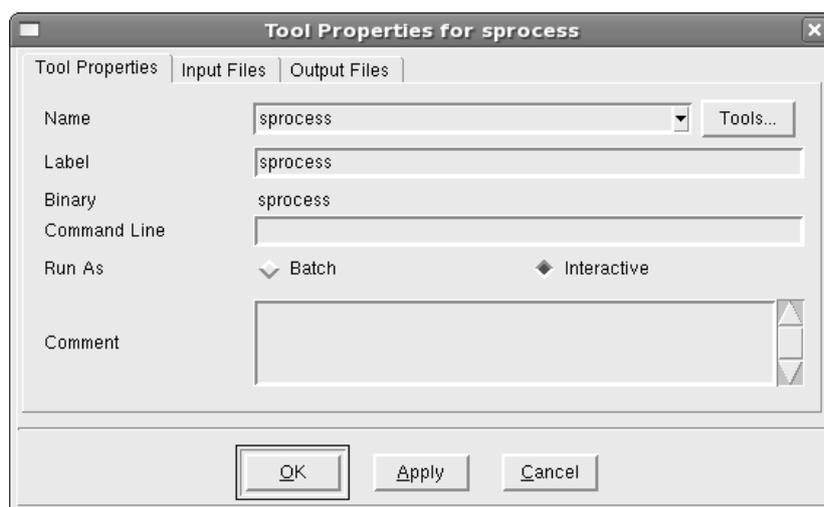


Рис. 3.1. Окно выбора режима запуска программы **SProcess**

Графический режим **Interactive** используется в демонстрационных целях, а также для отладки технологии и топологии структуры. Фоновый режим **Batch** требует меньше ресурсов компьютера, поэтому используется при расчете больших проектов. Все необходимые численные данные при

этом можно просмотреть в окне протокола. Для этого необходимо выбрать нужный, уже рассчитанный узел проекта и вывести его контекстное меню, в котором выбрать *View output*.

3.2. Структура командного файла модуля SProcess

Основной способ работы в **SProcess** заключается в написании текстового командного файла, содержащего последовательность команд, и запуск **SProcess** с этим файлом. Для создания командного файла необходимо выбрать **SProcess** в окне проекта, вызвать его контекстное меню и выбрать *Edit Input -> Commands*. После записи всех необходимых команд, файл необходимо сохранить, при этом **Workbench** автоматически присвоит файлу нужное имя.

SProcess не предъявляет жестких требований к последовательности команд, за исключением естественных ограничений, таких как требование определять все переменные до того, как они будут использованы. С целью облегчения чтения и редактирования, предпочтительным является разбивать командный файл на подразделы.

Рекомендованная структура командного файла имеет следующие подразделы:

- подраздел определения переменных;
- подраздел настройки используемых моделей;
- подраздел настройки сетки;
- подраздел определения фоторезистивных масок;
- подраздел последовательности технологических операций;
- подраздел преобразования и сохранения структуры.

3.3. Основные команды модуля SProcess и их параметры

В данном разделе приведены описания основных команд программного модуля **SProcess**, используемых при моделировании изготовления полупроводниковых кремниевых приборов. Команды указаны в порядке их использования в примере командного файла, приведенного в раздел 3.4. Следует учитывать, что интерпретатор **SProcess** чувствителен к регистру, т.е. если переменная имеет имя Nd, то обращение nd вызовет ошибку. **SProcess** игнорирует строки, начинающиеся со знака #. Сочетание ;# воспринимается как начало комментария, а концом комментария является конец строки. Знак \ означает разрыв команды с переходом на следующую строку. Т.е. команда

```
Init Silicon field=Boron resistivity=0.005  
wafer.orient=100 slice.angle=-90
```

вызовет ошибку, а команда

```
init field=Boron Silicon wafer.orient=100 \  
resistivity=0.005 slice.angle=-90
```

будет прочитана и выполнена нормально. Большинство команд **SProcess** допускают произвольный порядок параметров в командной строке.

3.3.1. Подраздел определения переменных

Объявление локальных параметров (переменных)

```
set [имя_переменной][значение_переменной]
```

Команда `set` служит для объявления локальных параметров (переменных) в командном файле модуля **SProcess**. В качестве значений параметра можно задать: число; строку символов либо одиночный символ, для этого они должны быть заключены в апострофы, например 'Constant'; параметр, объявленный в оболочке **Workbench**; уже определенный внутренний параметр **Sprocess**; выражение, содержащее различные варианты перечисленных величин. Обрамляющие символы `@` означают ссылку на имя параметра из строки параметров проекта в **Workbench**. Для каждого эксперимента **Workbench** подставляет значения параметров из соответствующей «ветки». Как и любой программный модуль, **SProcess** может оперировать только с теми внешними параметрами, которые были объявлены либо в нем, либо в предыдущем модуле. Символ `$` означает ссылку на внутренний параметр **SProcess**. Чтобы задать в качестве значения переменной выражение, требуется использовать отдельную команду:

```
[expr [операнд_1] [операция] [операнд_2] ...]
```

Пример:

```
Set N [expr $P/1e10+@Nd@-@Na@]
```

3.3.2. Подраздел настройки используемых моделей

Изменение системы координат

```
math coord.ucs
```

Выполняется изменение системы координат: ось *Y* направляется слева направо вдоль структуры; ось *X* – вглубь. Рекомендуется для лучшего согласования с другими программами пакета **SENTAURUS TCAD**.

Обновление графического отображения структуры

graphics on

Команда включает обновление графического отображения структуры при каждом ее изменении. Для наблюдения изменений в реальном времени необходимо выбрать графический режим в настройках модуля **SProcess**. Структура отображается при помощи модуля **Tecplot SV**.

Выбор режима контроля дозы при ионном легировании

pdbSet ImplantData DoseControl {WaferDose | BeamDose}

Режим *WaferDose* означает, что указанная в команде *implant* доза полностью попадет в подложку. В этом режиме итоговая доза легирования не зависит от того, под каким углом происходит легирование. *BeamDose* связывает указываемую дозу с концентрацией ионов примеси в легирующем луче. В этом случае итоговая доза может меняться в зависимости от таких параметров как *tilt* и *rotation*.

Задание размера сетки при формировании пленок нового материала

dbSet <материал> Grid perp.add.dist <значение>

Команда задает размер сетки при формировании пленок нового материала при нанесении, окислении и т.д. . *Значение* указывает на максимальное расстояние между соседними узлами сетки в выращиваемом материале (по умолчанию в сантиметрах). Если это расстояние превышает, между этими точками добавляется дополнительный узел.

Настройка модели ионной имплантации

implant species=<примесь> <материал> [imp.table=<файл> | \ tables=<имя>] [модель] [damage]

Команда *implant* используется как для настройки, так и для самого процесса ионной имплантации. На настройку указывают ключевые слова *species* или *tables*: *species= <примесь>* – тип ионов легирующей примеси *Во-*

ron, *Phosphorus*, *Arsenic* и т.д.; *material* – легируемый материал *Silicon*; *imp.table=<файл>* – имя файла, содержащего таблицы значений параметров ионного легирования, такие как средний нормальный пробег и среднеквадратичное отклонение для разных энергий легирования. Для того чтобы заменить все таблицы значений параметров распределений для какой-либо примеси во всех материалах используется ключевое слово *tables=<имя>*, *material* при этом опускается. В качестве *имени* можно задать один из вариантов: *Default*, *Taurus*, *Tasch*, *AdvCal*, *Dios*, *TSuprem4*. При выборе следует учитывать, что некоторые из этих таблиц, в частности *Taurus*, также меняют и другие настройки ионного легирования. Параметр *модель* меняет функцию распределения примеси. Доступны следующие варианты:

- gaussian* – нормальное распределение Гаусса;
- pearson* – распределение Пирсон-4 (по умолчанию);
- pearson.s* – распределение Пирсон-4 с учётом экспоненциального «хвоста»;
- dualpearson* – двойное распределение Пирсона;
- point.response* – функция распределения, основанная на заданных внешних аналитических данных. Доступна только в одномерном случае.

Команда *damage!/damage* включает/выключает расчет радиационных повреждений структуры при ионном легировании.

Примеры:

```
implant species=Boron Silicon imp.table=my_table.tab \  
pearson !damage
```

```
implant species=Boron tables=Dios
```

```
implant tables=Taurus
```

Настройка модели диффузии примеси

```
pdbSet <material> Dopant DiffModel <модель>
```

В команде модели диффузии указываются: *material* – материал, в котором происходит диффузия; *модель* – имя используемой модели диффузии. **SProcess** поддерживает следующие модели переноса легирующих примесей и точечных дефектов в процессе диффузии:

ChargedPair – трёхпоточная модель многочастичной диффузии, в которой рассматриваются парные взаимодействия вида $AZ+IC$ $AI(Z+C)$ и

$AZ+VC$ $AV(Z+C)$ между атомами легирующей примеси A , междоузлиями I и вакансиями V с учетом их зарядовых состояний Z для примеси и C для точечных дефектов путем совместного решения системы из трех диффузионных уравнений, определяющих движение каждого типа частиц в отдельности с учетом описанных выше взаимодействий. Она представляет собой хороший вариант соблюдения баланса между вычислительными затратами и точностью расчета;

ChargedReact – наиболее сложная и физически наиболее точная пяти-поточковая модель многочастичной диффузии, которая рассматривает взаимодействия между пятью типами частиц: атомами примеси A , междоузлиями I , вакансиями V и всевозможными их соединениями AV , AI с учетом зарядовых состояний всех частиц. Для каждой легирующей примеси рассматривается система из пяти дифференциальных уравнений с частными производными. Так как расчеты по данной модели связаны с большими затратами машинного времени, то ее рекомендуется использовать при экстремальных перепадах температуры для быстропротекающих процессов, при специфических начальных условиях и т. д.;

ChargedFermi – аналог модели в приближении эффективного коэффициента диффузии с учетом зарядовых состояний точечных дефектов, находящихся в тепловом равновесии. Данная модель может быть использована при длительных высокотемпературных операциях, где эффекты переходных процессов при отжиге послеимплантационных дефектов минимальны;

Pair – упрощенная модель *ChargedPair* на основе уравнений взаимодействия $A+I$, AI и $A+V$, AV без учета зарядовых состояний частиц;

React – упрощенная модель *ChargedReact*, не учитывающая в уравнениях взаимодействия частиц их зарядовых состояний;

Fermi – упрощенная модель *ChargedFermi*, тождественная приближению с эффективным коэффициентом диффузии без учета зарядовых состояний;

Constant – наиболее простая модель с постоянным коэффициентом диффузии. Здесь не учитывается взаимодействие между легирующей примесью и точечными дефектами, а также отсутствует эффект влияния электрического поля на движение примеси. Эта модель применяется главным образом для описания диффузии окислителя в окисле.

По умолчанию используется *ChargedPair*.

Пример:

```
pdbSet Silicon Dopant DiffModel ChargedReact
```

3.3.3. Подраздел настройки сетки

Отличительной особенностью **SProcess** является возможность гибкой настройки сетки. Все многообразие возможностей построения сеток в **SProcess** можно свести к двум типам сеток: адаптивным и неадаптивным (статичным). Основным средством задания сеток обоих типов служит команда `refinebox`.

Команда задания сетки на определенной области структуры

```
refinebox min= {x1 y1 z1} max= {x2 y2 z2} \  
xrefine= {размеры_элемента} yrefine= {размеры_элемента} \  
zrefine = {размеры_элемента} [материал] [регион]
```

Геометрические размеры области определяются параметрами `min` и `max`, при этом возможны 1D-, 2D- и 3D-случай. Параметры `xrefine`, `yrefine` и `zrefine` задают размеры элемента сетки по соответствующей оси координат. Если для оси указаны три значения, размер элемента является квадратичной функцией, два значения – линейной от меньшего размера к большему, одно значение – размер элемента сетки постоянен. В качестве области определения `refinebox` может быть задан конкретный материал или регион, для этого его название должно быть добавлено в командную строку. В случае если указан и материал/регион, и `min`, `max`, то `refinebox` определяется в указанной области, но только для определенного материала/региона.

Примеры:

```
refinebox min= {-0.25 0.4 0.0} max= {0.4 0.6 1.0} \  
xrefine= {0.1 0.06 0.1} yrefine= {0.1 0.01 0.1} \  
zrefine = {0.01} silicon
```

```
refinebox min= {0.6 0.6} max= {0.8 0.8} xrefine= {0.1} \  
yrefine= {0.1 0.01}
```

```
refinebox oxide xrefine= {0.2} yrefine= {0.2}
```

Удаление всех заданных областей определения сетки

```
refinebox clear
```

Данную команду рекомендуется использовать в начале моделирования для переопределения глобальных настроек сетки.

Обновление сетки во всей структуре

```
grid remesh
```

Команда обеспечивает принудительное обновление (перерасчет) сетки во всей структуре.

Адаптивная сетка отличается от статической тем, что она автоматически обновляется при каком-либо изменении структуры, например травлении, осаждении, диффузии и т.д. При этом есть возможность настроить как частоту обновлений, например, через каждые десять диффузионных шагов, так и условий, при которых будет произведен перерасчет сетки по выбранным критериям. Доступны следующие критерии: Relative difference (выбран по умолчанию), Absolute difference, Logarithmic difference, Inverse hyperbolic sine (asinh) difference, Gradient, Local dose error, Interval refinement. Каждый критерий имеет свои параметры, позволяющие менять его условия обеспечения точности, и тем самым регулировать размер элементов сетки. Для добавления критерия перерасчета сетки следует задать его параметры либо глобально, либо для конкретной области, после чего при проверке сетки **SProcess** автоматически будет учитывать данный критерий. Чтобы исключить тот или иной критерий следует снизить порог его требований до заведомо выполнимых.

Глобальное задание параметров критериев обеспечения точности

`pdbSet Grid <поле> <параметр> <значение>`

Команда глобального задания параметров критериев включает следующие параметры: *поле* – имя поля, для которого проводится проверка соответствия критерию, например, концентрации примеси; *параметр* – имя параметра данного критерия, например, для Relative difference доступны Refine.Abs.Error и Refine.Rel.Error; *значение* – собственно значение данного параметра. Если указать ключевое слово AdaptiveField, то критерий установится для всех доступных полей.

Примеры:

```
pdbSet Grid AdaptiveField Refine.Abs.Error 1e25
```

```
pdbSet Grid AdaptiveField Refine.Rel.Error 2.0
```

Команда задает глобальное значение для допустимой абсолютной и относительной разницы значений для всех доступных полей в соседних узлах сетки. Так как глобальные параметры определены для всей структуры, разумно выбирать их наименее жесткими, т.е. наибольшими по значению.

Примеры:

```
pdbSet Grid Damage Refine.Min.Value 1e25
```

```
pdbSet Grid Damage Refine.Max.Value 1e25
```

```
pdbSet Grid Damage Refine.Target.Length 1
```

Здесь же задаются глобальные параметры критерия Interval refinement для повреждений структуры Damage, появляющихся в результате ионной имплантации.

Задание локальной адаптивной сетки

```
refinebox name=<имя> min= {x1 y1 z1} max= {x2 y2 z2} \  
Adaptive refine.fields= {имя} <параметр>= {значение} \  
[refine.min.edge= {размеры_элемента}] \  
[refine.max.edge= {размеры_элемента}] [материал] [регион]
```

Для задания адаптивной сетки в команде refinebox указываются: name=<имя> – идентификатор самой области определения сетки, который задается произвольно; refine.fields={имя} – идентификаторы полей, значения которых должны удовлетворять критериям перерасчета сетки; Adaptive – ключевое слово, указывающее, что данная команда refinebox задает адаптивную сетку; <параметр>= {значение} – параметр критерия перерасчета сетки; refine.min.edge/refine.max.edge – минимальные/максимальные разрешенные размеры элемента сетки, значения которых задаются для разных осей в порядке X, Y, Z. Прочие параметры идентичны случаю статической сетки.

Примеры:

```
refinebox name=P_well_box Silicon Adaptive
```

```
refine.fields= {Boron Phosphorus Arsenic} \  
rel.error= {Boron=0.5 Phosphorus=0.5 Arsenic=0.5} \  
abs.error= {Boron=1e15 Phosphorus=1e15 Arsenic=1e15} \  
min= {0 $Y1_well} max= {1 $Y2_well} \  
refine.min.edge= {0.05 0.1}
```

Включение адаптивной сетки

```
pdbSet Grid Adaptive 1
```

Включение учета заданных пользователем линий

```
#pdbSet Grid SnMesh UseLines 1
```

Команда обеспечивает учет заданных пользователем линий (команда line) при построении сетки. Рекомендуется при использовании адаптивной сетки.

Задание количества шагов, после которых проводится проверка сетки на соответствие существующим критериям

```
pdbSet Diffuse Compute.Regrid.Steps <значение>  
pdbSet Diffuse Growth.Regrid.Steps <значение>  
pdbSet Diffuse Epi.Regrid.Steps <значение>
```

Задается количество шагов, после которых проводится проверка сетки на соответствие существующим критериям: Compute.Regrid.Steps – количество шагов во время диффузии без окисления; Growth.Regrid.Steps – количество шагов при окислении и силицидизации; Epi.Regrid.Steps – количество шагов при эпитаксиальном наращивании. Если значение выбрать отрицательным, то при данном процессе не будет выполняться проверка сетки.

Задание допустимой доли узлов сетки, в которых не выполняются условия проверки

```
pdbSet Grid Refine.Percent <значение>
```

Процесс перестройки сетки не производится при единичных несоответствиях критериям. Эта команда позволяет задать долю узлов сетки, в которых не выполняются условия проверки, чтобы запустить перерасчет сетки.

3.3.4. Подраздел определения фоторезистивных масок

Задание масок для создания структуры

```
mask name=<имя> (left=<значение> right=<значение> \  
front=<значение> back=<значение> | \  
segments=<список_значений>) [negative]
```

Команда задает маску, которую потом можно использовать при создании структуры: name=<имя> – произвольный идентификатор маски; left, right, front, back – координаты краев маски в микрометрах, при этом для 2D-случая требуется указать лишь left и right. В случае сложной маски допускается последовательное объявление нескольких команд mask с одинаковым именем и разными координатами. segments – альтернативный способ зада-

ния положения маски, который удобен при создании сложных масок. При такой записи для описания сегмента маски используются два последовательно расположенных числа, обозначающих, соответственно, левый и правый края маски. Следующие два числа – координаты второго сегмента и так далее. *negative* – инверсия маски.

Примеры:

```
mask name=Contact_mask left=-1 right=$Y1_SC
```

```
mask name=Contact_mask left=$Y2_SC right=$Y1_Gate+@L_Gate@/2
```

```
mask name=mask1 segments={1.65 0.15 1.95 0.6} negative
```

3.3.5. Подраздел последовательности технологических операций

Задание линии для построения сетки

```
line <ось> location=<координата> [spacing=<значение>] / [tag=<имя>]
```

Команда определяет линии, которые в дальнейшем могут быть использованы при построении сетки: *ось* – имя оси, вдоль которой создается линия; *location=<координата>* (возможно сокращение *loc*) – определяет положение линии по выбранной оси (в микрометрах); *spacing=<значение>* (возможно сокращение *sra*) – изначальный размер сетки в микрометрах; *tag=<имя>* – задает имя линии, на которое в дальнейшем можно ссылаться, в частности, в команде *region*. Рекомендуется создавать линии по краям используемых фоторезистивных масок.

Примеры:

```
line x location=0 spacing=1 tag=top
```

```
line x loc=100 spa=1 tag=botton
```

```
line y location=30<nm> spacing=10<nm>
```

Определение параметров области подложки

```
region <материал> [xlo=<линия>] [ylo=<линия>] [zlo=<линия>] /
```

```
[xhi=<линия>] [yhi=<линия>] [zhi=<линия>] [name=<имя>] /
```

```
[min= {x1 y1 z1} max= {x2 y2 z2}] [field=<поле> /
```

```
(resistivity=<значение> | concentration=<значение>)]
```

Команда определяет регион, который может быть использован как подложка. Местоположение региона может быть задано несколькими способами. Первый способ – указание линий, которыми ограничен объявляемый регион. Для этого в полях *xlo/xhi* и т.д. следует задать указанные в *tag*

идентификатор нужных линий. Если часть либо все поля xlo/xhi/ylo/yhi/zlo/zhi опущены, границами региона принимаются самые крайние линии. Второй способ аналогично команде refinebox обеспечивает задание региона непосредственно через координаты при помощи min и max. В команде указываются следующие параметры: *материал* – материал, которому присваивается регион; name=<имя> – присвоение региону имени; concentration=<значение> – определение концентрации в см⁻³ в регионе, при этом при помощи параметра field=<поле> должно быть указано поле, концентрация которого определяется, например, примесь бора; resistivity=<значение> – альтернативная форма задания концентрации поля через удельное сопротивление, которое может быть указано только для случая, когда объявленное поле – электрически активная примесь.

Примеры:

```
region name=bulk min= {-5.0 0.0 0.0} max= {5.0 1.0 1.0}
```

```
region poly ylo=left yhi=right xlo=surf xhi=Gate
```

```
region silicon xlo=bottom xhi=top field=Boron \  
concentration=1E15
```

```
region Silicon
```

Инициализация полупроводниковой подложки

```
init <материал> [field=<поле> (resistivity=<значение> | \  
concentration=<значение>)] wafer.orient=<ориентация> \  
slice.angle=<угол>
```

Описание параметров полупроводниковой подложки: *материал* – материал подложки; wafer.orient=<ориентация> – кристаллографическая ориентация подложки, например, 100; slice.angle=<угол> – угол между системой координат подложки и системой координат **SProcess** (по умолчанию -90°). Остальные параметры аналогичны тем, что и в команде region.

Пример:

```
init Silicon field=Boron resistivity=0.005 \  
wafer.orient=100 slice.angle=-90
```

Осаждение выбранного материала

```
deposit <материал> type=<тип> [thickness=<значение>] /  
[coord=<координата>] [time=<время> (rate=<скорость>)] /  
[поле] [concentration=<значение>]
```

Описание операции осаждение выбранного *материала*: *type*=<тип> – тип осаждения: *isotropic* – равномерное осаждение во всех направлениях; *anisotropic* – осаждение строго в вертикальном направлении; *fill* – заполнение всей структуры выбранным материалом до определенного уровня; *fourier* – тип осаждения, позволяющий задать направления нарастания осаждаемого слоя; *crystal* – осаждение со скоростью, зависящей от кристаллографической ориентации; *thickness* = <значение> – определение толщины осаждаемого слоя в микрометрах). Альтернативно может быть задана скорость *rate* и время осаждения *time*. В случае режима *fill* указывается координата *coord*, до которой структура будет заполнена выбранным материалом. Также можно задать тип изначальной примеси или другое поле в осаждаемом материале, для этого требуется указать название *поля*. Параметр *concentration* = <значение> определяет концентрацию выбранного поля в см⁻³.

Примеры:

```
deposit nitride type=isotropic thickness=160<nm>
deposit anisotropic nitride time=10<min> rate=10<nm/s>

deposit material=oxide type=fill coord=-0.4

deposit poly type=isotropic thickness=0.2<um> \
Phosphorus concentration=1e19<cm-3>
```

Нанесение фоторезистивной маски

```
photo mask=<имя> thickness=<значение>
```

В команде *photo* нанесения фоторезистивной маски указываются: *mask*=<имя> – имя ранее определенной маски; *thickness*=<значение> – толщина фоторезистивного слоя. Для удаления маски служит команда *strip resist*. Также можно удалять и другие материалы. Следует отметить, что для таких команд как *deposit*, *etch*, *implant* есть возможность задавать маску прямо в теле команды. В таком случае физически фоторезист не наносится, и команда *strip* не требуется. Однако следует учитывать, что в этом случае параметры *left* и *right*, указываемые в команде *mask*, воспринимаются как края окон в фоторезисте, а не самой маски. Таким образом, маска при объявлении в теле команды инвертируется. Для устранения данного эффекта может быть использовано ключевое слово *negative*.

Примеры:

```
mask name=mask1 left=0 right=10
mask name=mask2 left=0 right=10 negative
```

```
;/# с использованием команды photo  
photo mask=mask1 thickness=2<um>  
deposit Oxide isotropic thickness=100<nm>  
strip resist
```

```
;/# с заданием маски в теле программы  
deposit Oxide isotropic thickness=100<nm> mask=mask2
```

Травление указанного материала

```
etch <материал> type=<тип> [thickness=<значение>] /  
[coord=<координата>] [time=<время> (rate=<скорость>)] /  
etchstop=<материал>
```

Синтаксически команда травления указанного *материала* очень схожа с командой deposit. Вместо режима fill в команде etch может быть указан тип травления cmp. Этот режим травления моделирует химико-механическую полировку и удаляет всю структуру независимо от материала и расположения до заданного параметром coord уровня. Помимо толщины стравливаемого слоя и времени травления можно также указать стоп-слой etchstop, по достижению которого в какой-либо точке процесс травления останавливается.

Примеры:

```
etch nitride type=anisotropic thickness=200<nm>  
etch type=cmp coord=0<nm>  
etch oxide type=isotropic etchstop=silicon  
etch silicon isotropic thickness=0.3 mask=Trench_mask
```

Ионная имплантация выбранной примеси

```
implant <примесь> dose=<доза> energy=<значение> \  
[tilt=<угол>] [mult.rot=<число>] [rotation=<угол>]
```

В команде, описывающей процесс ионной имплантации, задаются следующие параметры: dose=<значение> – доза имплантируемой примеси, по умолчанию в см⁻²; energy=<значение> – энергия легирования, которая указывается в эВ, кэВ, МэВ; tilt=<угол> – угол между структурой и ионным лучом ion beam в градусах с учетом системы координат $X_w Y_w Z_w$, связанной с пластиной (рис. 3.2). Также можно выбрать режим мультилегирования, для чего требуется задать параметр mult.rot=<число>. В этом режиме легиро-

вание проводится в несколько этапов, число которых равно значению параметра *число*, с дозой равной *доза/число*. После каждого шага легирования структура поворачивается в своей плоскости на угол $360/\text{число}$. Начальный угол задает параметр *rotation*=<угол> (по умолчанию -90°).

Следует отметить, что в системе координат при моделировании в модуле **Sprocess** ось X_S направлена в глубину пластины, а оси Y_S и Z_S лежат в плоскости пластины. Угол между осями Y_W и Y_S *slice.angle* по умолчанию принимается равным -90° .

Примеры:

```
implant Arsenic dose=3.5e15<cm-2> energy=1.5<MeV> mult.rot=4
```

```
implant Boron dose=1e13 energy=80 tilt=7 rotation=0 \
mask=P_plus_mask
```

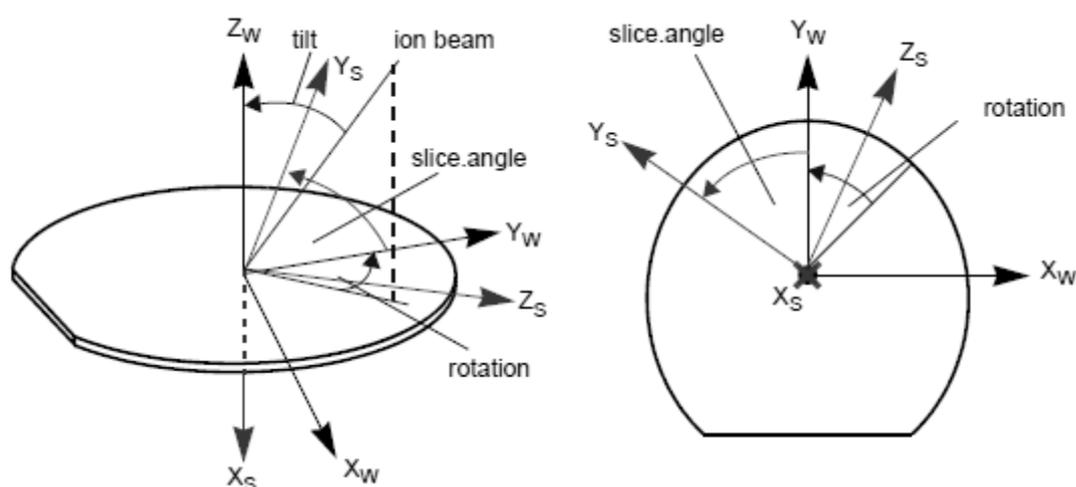


Рис. 3.2. Определение направления ионного пучка в системе координат $X_W Y_W Z_W$, связанной с пластиной

Команда описания высокотемпературных операций

```
diffuse temperature=<значение> time=<время> \
[temp.ramp=<имя>] [<атмосфера> | [gas_flow=<имя> | \
flow<атмосфера>=<значение> | p<атмосфера>=<значение> | \
partial.pressure= {<атмосфера1>=<значение> \
<атмосфера2>=<значение> ...} | flows= \
{<атмосфера1>=<значение> <атмосфера2>=<значение> ...}] \
[epi.doping= {<поле1>=<значение> <поле2>=<значение> ...}] \
[epi.thickness=<значение>]
```

Универсальная команда, позволяющая выполнять высокотемпературные операции, такие как диффузионная разгонка, окисление и эпитаксиаль-

ное наращивание. Помимо температуры `temperature` и времени `time` проведения операции важным параметром является заданная атмосфера, в которой проводится процесс, так как она определяет выбор осуществляемой операции. Доступны опции: O2, H2O, N2, H2, HCl, Eri. В последнем случае осуществляется эпитаксиальное наращивание. Для выращенного слоя можно указать толщину `eri.thickness` (может быть сокращено до `thick`) и концентрацию исходных примесей `eri.doping`. Для выбранной атмосферы можно указать давление `p<атмосфера>=<значение>` в атмосферах, либо поток газа при помощи `flow<атмосфера>=<значение>` в л/мин. Также есть возможность задать сложную атмосферу, состоящую из нескольких компонент. В этом случае для каждого компонента нужно указать либо парциальное давление `partial.pressure= {<атмосфера1>=<значение> <атмосфера2>=<значение> ...}`, либо поток `flows= {<атмосфера1>=<значение> <атмосфера2>=<значение> ...}`. Кроме того, атмосфера может быть определена отдельно командой `gas_flow` и затем указана в конкретной операции при помощи `gas_flow=<имя>`. В этом случае появляется возможность использовать одну атмосферу в нескольких процессах без необходимости заново ее переопределять. Синтаксис `gas_flow` аналогичен заданию атмосферы в команде `diffuse`:

```
gas_flow name=<имя> [<атмосфера> | flow<атмосфера>= \
<значение> | p<атмосфера>=<значение> | partial.pressure= \
{<атмосфера1>=<значение> <атмосфера2>=<значение> ...} | \
flows= {<атмосфера1>=<значение> <атмосфера2>=<значение> ...}]
```

Схожа с командой `gas_flow` команда `temp_ramp`, которая позволяет проводить диффузионные операции в несколько стадий с различными скоростями набора и снижения температуры:

```
temp_ramp name=<имя> temperature=<значение> time=<время> \
ramprate=<значение> [<атмосфера>] last
```

Параметр `ramprate=<значение>` определяет в зависимости от знака скорость подъема либо спада температуры, по умолчанию значение указывается в градусах в минуту. Атмосфера в команде `temp_ramp` может быть задана всеми теми же способами, что и в команде `diffuse`. Для использования объявленного `temp_ramp` в команде `diffuse` следует указать `temp.ramp=<имя>`. Для создания многостадийного процесса команда `temp_ramp` может быть указана несколько раз с одинаковым аргументом `name=<имя>`. В этом случае при объявлении в `diffuse` режимы, указанные в `temp_ramp`, будут выполняться последовательно. Ключевое слово `last` указывает на завершение цикла `temp_ramp`.

Примеры:

```
diffuse temperature=900<C> time=10<s> H2O

diffuse temperature=950<C> time=28<min> \
flows= {H2O=0.75<l/min> N2=2.2<l/min>}

diffuse temperature=800<C> time=60<min> Epi thick=0.01 \
epi.doping = {Germanium=8e21<cm-3> Boron=1e15<cm-3>}

temp_ramp name=MyCycle temperature=500 time=5< H2O \
ramprate=100<C/min>

temp_ramp name=MyCycle temperature=1000 time=10 O2

temp_ramp name=MyCycle temperature=1000 time=10 N2 \
ramprate=-50<C/min> last

diffuse temp.ramp=MyCycle

gas_flow name=MyGasFlow pH2O=0.5 pO2=0.5 pH2=0.1

temp_ramp name=MyTempRamp temperature=1000<C> time=10<min> \
gas.flow=MyGasFlow last

diffuse temp.ramp=MyTempRamp
```

3.3.6. Подраздел преобразования и сохранения структуры

Если моделируемая структура обладает осью симметрии, например, элементарный MOS-транзистор, то в таких случаях имеет смысл проводить моделирование лишь половины структуры с последующим ее зеркальным отображением относительно этой оси. Для этого используется команда `transform`.

Универсальная команда преобразования структуры

```
transform <тип> [location=<координата>] [направление] \
[translate= {вектор}] [axis=<ось>] [angle=<угол>] \
length=<значение> [min= {x1 y1 z1} max= {x2 y2 z2}] [!keep.original]
```

В качестве параметра *тип* можно задать: `reflect` – зеркальное отражение; `stretch` – растягивание в выбранном направлении; `cut` – обрезание; `flip` – переворачивание; `rotate` – поворот; `translate` – перемещение. По умолчанию трансформация производится над всей структурой, однако можно локализовать ее действие на ограниченную область, указав ее координаты при помощи полей `min/max`. `location=<координата>` указывает точку, относительно которой проводится преобразование. Ось, на которой лежит эта точка,

определяется автоматически при задании направления трансформации: left и right – ось X, up и down – ось Y, front и back – ось Z. Если выбран параметр stretch, то необходимо также задать параметр length=<значение>, указывающий на сколько требуется растянуть структуру в микрометрах. Для параметра rotate, который доступен только в 3D, требуется определить ось axis=<ось> и угол поворота angle=<угол>. При этом доступны углы 90°, 180° и 270°. При задании режима translate сразу за объявлением типа преобразования должен быть указан вектор переноса, составляющие которого по умолчанию считаются в микрометрах.

По умолчанию все перечисленные преобразования сохраняют исходную структуру. Чтобы оставить только преобразованную структуру требуется использовать ключевое слово !keep.original.

Примеры:

```
transform reflect right
```

```
transform stretch location=0.7 length=1.5 left
```

```
transform cut location=1.0 down
```

```
transform cut min= {0<um> 0<um>} max= {1<um> 3<um>}
```

```
transform translate = {-1 0 0}
```

```
transform flip
```

```
transform rotate axis=X angle=90
```

Создание контакта к структуре

```
contact name=<имя> <тип> ([xlo=<координата> \  
ylo=<координата> zlo=<координата> xhi=<координата> \  
yhi=<координата> zhi=<координата>] | [x=<координата> \  
y=<координата> z=<координата>]) <материал> [region=<регион>]
```

В модуле **SProcess** существует два *типа* контактов: box и point. Первый определяет контактом поверхность структуры, попадающую в заданную прямоугольную зону. Эта зона задается полями xlo/xhi/ylo/yhi/zlo/zhi. В случае box-контакта требуется указать материал либо регион, которому принадлежит контакт. Контактom типа point становится весь *регион*, которому принадлежит указанная точка, задается через x/y/z. Еще одним вариантом определения контакта является объявление контактом всей обратной стороны структуры, и для этого вместо координат указывается ключевое слово bottom. Указание *типа* в данном случае не требуется.

Примеры:

```
contact name= "substrate" bottom
contact name=source box Silicon xlo= 0.0 xhi= 0.005 \
ylo= -0.4 yhi= -0.2
contact name=gate x=-0.05 y=0.0 point
```

Сохранение структуры в формате TDR

```
struct tdr=<ИМЯ>
```

При сохранении структуры в формате TDR итоговое имя файла будет <ИМЯ>_fps.tdr.

Пример:

```
struct tdr=n@node@
```

Сохранение структуры в двух форматах

```
struct smesh=<ИМЯ>
```

Эта команда обеспечивает сохранение структуры в двух форматах: fps.tdr – содержит сетку и распределения примесей и различных физических полей; bnd.tdr – содержит сетку с распределениями примесей и различных физических полей.

Пример:

```
struct smesh=output
```

Такими же способами можно сохранять не только готовую структуру, но и промежуточные стадии.

3.4. Пример командного файла модуля SProcess

В данной работе рассмотрен пример планарной технологии изготовления нормально закрытого *n*-MOS-транзистора с LDD-областями (*Lightly Doped Drain*) (рис. 3.3), формирование которого с помощью командного файла (листинг 3.1) в модуле **SProcess** реализуется с помощью заданной последовательности технологических операций.

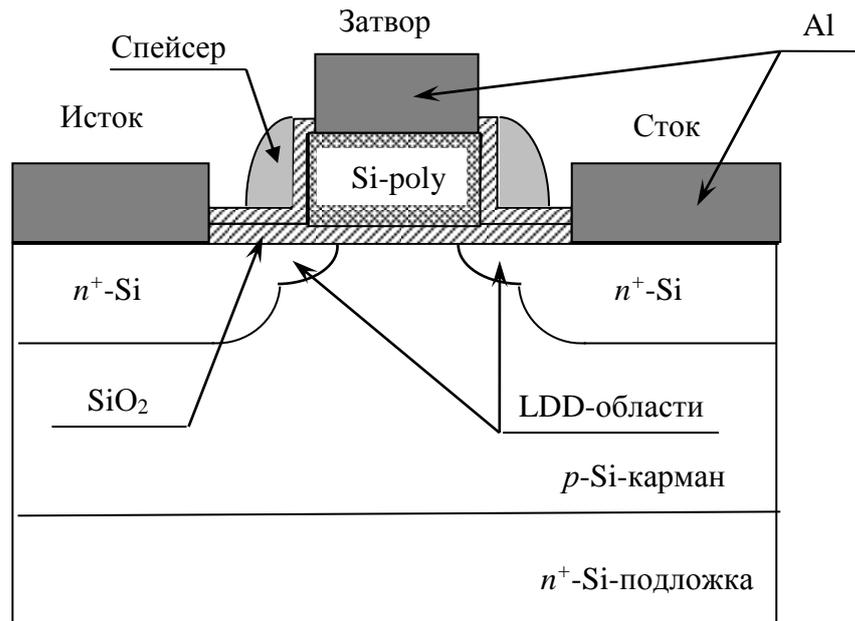


Рис. 3.3. Структура нормально закрытого *n*-MOS-транзистора

1. Исходная кремниевая подложка марки КЭФ4,5 с ориентацией (100) .
2. Многоэтапная имплантация бора для создания *p*-кармана: с энергией 200 кэВ и дозой $2,0 \cdot 10^{13} \text{ см}^{-2}$; с энергией 100 кэВ и дозой $1,0 \cdot 10^{13} \text{ см}^{-2}$; с энергией 25 кэВ и дозой $3,0 \cdot 10^{12} \text{ см}^{-2}$.
3. Активационный отжиг ионно-имплантированного бора в инертной атмосфере при температуре 1000 °С в течение 20 секунд.
4. Нарращивание тонкого подзатворного окисла во влажном кислороде при температуре 900 °С в течение 20 минут.
5. Осаждение поликремния с уровнем легирования $1,0 \cdot 10^{19} \text{ см}^{-3}$ толщиной 0,2 мкм для последующего создания поликремниевого затвора.
6. Нанесение фоторезистивной маски для формирования поликремниевого затвора при помощи фотолитографии.
7. Анизотропное травление поликремния.
8. Окисление поликремния при температуре 1000 °С в течение 20 секунд в сухом кислороде.
9. Создание слабелегированных LDD-областей стока и истока легированием мышьяка с относительно небольшой дозой $1,0 \cdot 10^{14} \text{ см}^{-2}$ и энергией 10 кэВ с активационным отжигом при температуре 1000 °С в течение 10 секунд.

10. Нанесение окисла толщиной 20 нм и осаждение нитрида кремния толщиной 50 нм для последующего формирования спейсеров.
11. Формирование спейсеров анизотропным травлением нитрида кремния.
12. Ионная имплантация мышьяка с энергией 40 кэВ и дозой $3,0 \cdot 10^{15}$ см⁻² поликремния, стоковой и истоковой областей.
13. Диффузионная разгонка мышьяка в атмосфере сухого кислорода при температуре 1000 °С в течение 20 секунд.
14. Нанесение фоторезистивной маски для формирования контактных окон.
15. Вскрытие контактных окон и удаление фоторезиста.
16. Формирование алюминиевых контактов.

Листинг 3.1

```

#-----Определение внутренних параметров-----

set    L_N_plus    @L_N_plus@    ;# um
set    L_Gate     @L_Gate@     ;# um

set    Dose_LDD   1e14    ;# sm-2
set    E_LDD     10      ;# keV
set    Dose_N_plus 3e15    ;# sm-2
set    E_N_plus   40      ;# keV

set    X_max     5       ;# um
set    Y_max     [expr $L_Gate/2+$L_N_plus] ;# um
set    Gate_edge [expr $L_Gate/2] ;# um
set    Cont_edge [expr $Y_max-0.2] ;# um
set    Me_edge   [expr $Y_max-0.3] ;# um

#-----Глобальная настройка моделей-----

math coord.ucs
graphics on
pdbSet ImplantData DoseControl {WaferDose}
pdbSet Oxide Grid perp.add.dist 1e-7
pdbSet Poly Grid perp.add.dist 1e-8

;;/# выбор моделей ионной имплантации и диффузии
implant tables=Taurus
pdbSet Silicon Dopant DiffModel ChargedPair

```

#-----Настройка сетки-----

Глобальные настройки сетки:

```
pdbSet Grid Adaptive 1
pdbSet Grid SnMesh UseLines 1
pdbSet Grid AdaptiveField Refine.Abs.Error 1e25
pdbSet Grid AdaptiveField Refine.Rel.Error 2.0

pdbSet Grid Damage Refine.Min.Value 1e25
pdbSet Grid Damage Refine.Max.Value 1e25
pdbSet Grid Damage Refine.Target.Length 1

pdbSet Diffuse Compute.Regrid.Steps 10
pdbSet Diffuse Growth.Regrid.Steps 20
pdbSet Grid Refine.Percent 0.01
```

Задание областей определения сетки:

```
refinebox clear
refinebox oxide xrefine= {0.05 0.01} yrefine= {0.05 0.01}
refinebox poly xrefine= {0.10 0.01} yrefine= {0.10 0.01}
```

;*# кремниевая подложка*

```
refinebox name=Substrate_box Silicon Adaptive
refine.fields= {Phosphorus} \
rel.error= {Phosphorus=0.8} \
abs.error= {Phosphorus=1e16} \
min= {0.0 0.0} max= {$X_max $Y_max} \
refine.min.edge= {0.08 0.05} refine.max.edge= {0.5 0.3}
```

;*# p-карман*

```
refinebox name=P-well_box Silicon Adaptive
refine.fields= {Boron} \
rel.error= {Boron=0.5} \
abs.error= {Boron=1e15} \
min= {0.0 0.0} max= {1.3 $Y_max} \
refine.min.edge= {0.04 0.02} refine.max.edge= {0.2 0.1}
```

;*# подзатворная (канальная) область*

```
refinebox name=Chanel_box Silicon Adaptive
refine.fields= {Boron} \
rel.error= {Boron=0.2} \
abs.error= {Boron=1e14} \
min= {0.0 0.0} max= {0.05 $Gate_edge} \
refine.min.edge= {0.001 0.001} refine.max.edge= {0.01 0.01}
```

```

;# n+-области стока и истока
refinebox name=N_plus_box Silicon Adaptive
refine.fields= {NetActive} \
rel.error= {NetActive=0.9} \
abs.error= {NetActive=1e15} \
max.dose.error= {Arsenic=1e8} \
min= {0.0 $Gate_edge-0.03} max= {0.2 $Y_max} \
refine.min.edge= {0.01 0.005} refine.max.edge= {0.1 0.05}

#-----Фоторезистивные маски-----

mask name=Gate_mask left=-1 right=$Gate_edge
mask name=Contact_mask left=-1 right=$Cont_edge negative
mask name=Metal_mask left=-1 right=$Me_edge

#-----Создание структуры-----

### Задание параметров подложки:
;# Определение размера структуры
line x location= 0.0
line x location= $X_max
line y location= 0.0
line y location= $Y_max
;# задание физических параметров подложки
region Silicon
init Silicon field=Phosphorus resistivity=4.5 wafer.orient=100 slice.angle=-90

### Создание p-кармана:
;# ионное легирование бора
implant Boron dose=2.0e13<cm-2> energy=200<keV> tilt=7 rotation=0
implant Boron dose=1.0e13<cm-2> energy=100<keV> tilt=7 rotation=0
implant Boron dose=3.0e12<cm-2> energy= 25<keV> tilt=7 rotation=0
;# активационный отжиг бора
diffuse temperature=1000<C> time=20<s>

### Создание затвора:
;# наращивание подзатворного окисла
gas_flow name=Gate_ox flowH2O=0.75<l/min> flowN2=2.0<l/min>
diffuse temperature=900<C> time=20<min> gas_flow=Gate_ox
;# нанесение легированного поликремния
deposit poly type=isotropic thickness=0.2<um> Phosphorus \
concentration=1e19<cm-3>
;# травление поликремния и подзатворного окисла
etch poly type=anisotropic etchstop=oxide mask=Gate_mask

```

etch oxide type=anisotropic etchstop=silicon
;# создание пленки окисла на поликремниевом затворе
diffuse temperature=1000<C> time=20<s> O2

Создание слаболегированных LDD-областей стока и истока:
;# легирование мышьяка с небольшой дозой $1 \cdot 10^{14} \text{ см}^{-2}$ и энергией 10 кэВ
implant Arsenic dose=\$Dose_LDD energy=\$E_LDD tilt=7 rotation=0
;# активационный отжиг мышьяка при температуре 1000 °C
;# в течение 10 секунд
diffuse temperature=1000<C> time=10<s>

Создание спейсера из нитрида кремния:
deposit oxide type=isotropic thickness=20<nm>
deposit nitride type=isotropic thickness=50<nm>
etch nitride type=anisotropic thickness=80<nm>

Создание n⁺-областей стока/истока:
;# имплантация мышьяка с энергией 40 кэВ и дозой $3 \cdot 10^{15} \text{ см}^{-2}$
implant Arsenic dose=\$Dose_N_plus energy=\$E_N_plus tilt=7 rotation=0
;# активационный отжиг мышьяка при температуре 1000 °C
;# в течение 20 секунд
diffuse temperature=1000<C> time=20<s>

Вскрытие контактных окон:
photo mask=Contact_mask thickness=1<um>
etch oxide type=anisotropic thickness=30<nm>
strip resist

Создание алюминиевых контактов:
;# напыление алюминия до определенного уровня
deposit Aluminum type=fill coord=-0.2<um>
;# травление алюминия
photo mask=Metal_mask thickness=1<um>
etch Aluminum type=anisotropic thickness=0.5<um>
strip resist

Зеркальное отражение половины структуры влево:
transform reflect left

#-----Задание контактов и сохранение структуры-----

contact name= Source box Aluminum xlo=-0.3 xhi=-0.1 /
ylo=-1*\$Y_max yhi=-1*\$Me_edge
contact name= Drain box Aluminum xlo=-0.3 xhi=-0.1 /

4. МОДУЛЬ СТРУКТУРНОГО ПРОЕКТИРОВАНИЯ SENTAURUS STRUCTURE EDITOR

4.1. Назначение модуля SDE

Модуль **SENTAURUS Structure Editor (SDE)** в отличие от **SProcess** является средством структурного моделирования полупроводниковых приборов. Он позволяет абстрагироваться от непосредственных режимов технологических операций и удобен, когда сама технология создания различных областей и слоев исследуемой структуры не важна, зато необходима точность их геометрических размеров и взаиморасположения. Немалым преимуществом является также то, что расчет в **SDE** выполняется на порядок быстрее, чем **SProcess**. Благодаря этому **SENTAURUS Structure Editor** оказывается очень полезен при теоретическом изучении структур и их поведения при тех или иных значениях конструктивных параметров, а также первичном анализе разрабатываемой конструкции. Кроме того **SDE** может применяться в паре с **SProcess**, когда второй моделирует создание всех ключевых областей структуры, а первый – второстепенных, например, межслойных диэлектриков и слоев металлизации.

SDE имеет развитый графический интерфейс, позволяющий сделать процесс создания структурных моделей полупроводниковых приборов достаточно наглядным и в прямом смысле слова «нарисовать» модель прибора. Для вызова интерфейса, показанного на рисунке 4.1, достаточно запустить **SDE** в режиме *Interactive*, или выбрать *Edit Input -> Boundary file* в контекстном меню. Однако гораздо более гибким и удобным подходом является применение **SDE** в составе проекта **Workbench** и написание командных файлов, содержащих полное описание конструкции, топологии, технологических параметров структуры.

4.2. Структура командного файла модуля SDE

Командные файлы для **SENTAURUS Structure Editor** пишутся на специальном языке программирования, основанном на логическом языке *Scheme*. Создать командный файл можно, запустив **SDE** в окне проекта **Workbench** и выбрав в его контекстном меню *Edit Input -> Commands*.

Командный файл для модуля **SDE** состоит из следующих функциональных блоков:

- определение переменных;

- определение геометрических размеров и координат областей структуры;
- задание профилей распределения примесей и геометрическое размещение легированных областей структуры;
- определение контактов;
- задание правил построения расчетной сетки.

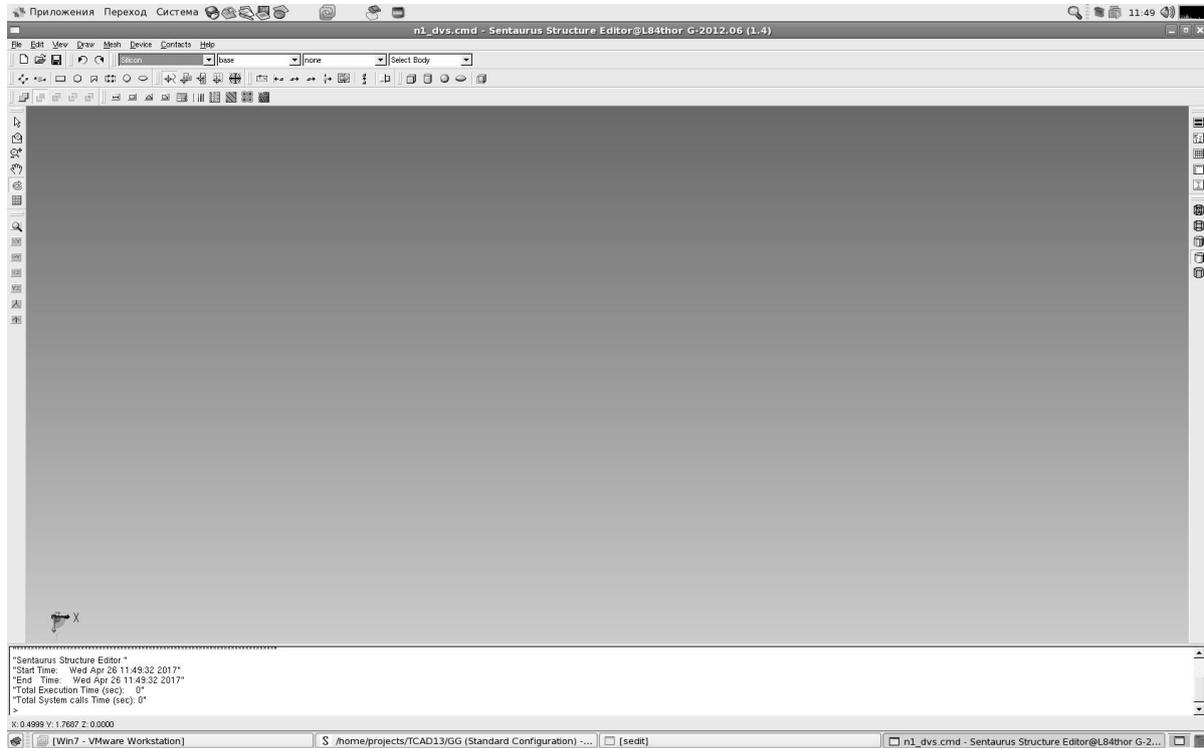


Рис. 4.1. Графический редактор SDE

4.3. Основные команды модуля SDE и их параметры

Ниже приведено описание основных команд модуля **SENTAURUS Structure Editor**, достаточных для создания элементарной модели полупроводникового прибора. Интерпретатор **SDE** чувствителен к регистру, т.е. если переменная имеет имя Nd, то обращение nd вызовет ошибку. Комментарием считается строка, начинающаяся с символа ; . В отличие от **SProcess** модуль **SDE** не нуждается в специальном знаке переноса команды на следующую строку: командой считается весь текст, расположенный между ключевым словом-именем текущей команды и началом следующей команды.

Таким образом, команда
`(sdegeo:create-rectangle (position 0 0 0) (position 10 10 0) "Silicon" "Substrate")`

может быть записана так:

```
(sdegeo:create-rectangle  
(position 0 0 0) (position 10 10 0) "Silicon" "Substrate")
```

или

```
(sdegeo:create-rectangle (position 0 0 0) (position 10 10 0)  
"Silicon"  
"Substrate")
```

или

```
(sdegeo:create-rectangle (position 0  
0 0) (position  
10 10 0) "Silicon" "Substrate")
```

и т.д.

4.3.1. Определение переменных

Объявление локальных параметров (переменных)

```
(define [имя_переменной] [значение_переменной])
```

Команда `define` позволяет объявлять локальные параметры в **SDE**. Значением параметра может быть: число; строка символов либо одиночный символ, которые должны быть заключены в апострофы, например, ‘Constant’; параметр, объявленный в оболочке **Workbench**; уже определенный внутренний параметр **SDE**; выражение, содержащее различные варианты перечисленных величин. Обрамляющие символы @ означают ссылку на имя параметра из строки параметров проекта. **SDE** может оперировать только с теми внешними параметрами, которые были объявлены либо в нем, либо в предыдущем модуле. В отличие **SProcess SDE** не нуждается в специальном символе для ссылки на ранее определенный внутренний параметр, следует указать только его имя. При задании в качестве значения переменной выражения, используется так называемая «польская запись»: знак операции предшествует операндам.

Примеры:

```
(define P_CONC_SUBS 5e20)  
(define XJ_LDD @Xj_LDD@)  
(define String ‘String’)  
(define Y_BOT_GATE (* -1 GATE_OX))  
(define Ne (- Nd Na))  
(define CONC_EPI (/ 1 (* @Rho_Epi@ (* 1.6e-19 450))))
```

4.3.2. Определение геометрических размеров и координат областей

Создание геометрических областей

```
(sdegeo:create-<тип> (position x1 y1 z1) (position x2 y2 z2) "материал" "имя")
```

Элементы структуры могут быть созданы одним из следующих геометрических примитивов: прямоугольник `rectangle`, многоугольник `polygon`, прямоугольный параллелепипед `cuboid` в трехмерном случае. Имеется возможность использовать цилиндры, сферы, пирамиды, торы, однако потребность в данных элементах возникает редко. Поля (`position ...`) задают размер и расположение примитива. Для прямоугольников и параллелепипедов указываются координаты двух диагональных углов. Для многоугольников необходимо указать координаты каждого угла. "материал" определяет материал, к которому будет отнесена создаваемая область, а "имя" задает идентификатор, используя который можно обращаться к области.

Примеры:

создание прямоугольника:

```
(isegeo:create-rectangle (position 0 0 0) (position 1 5 0) "Silicon" "Subctrate")
```

создание многоугольника:

```
(isegeo:create-polygon (list (position X1_Gate 0 0) (position X2_Gate 0 0) (position X2_Gate H_Gate 0) (position X1_Gate (+ H_Gate 0.5) 0) ) "PolySi" "PolyGate")
```

создание прямоугольного параллелепипеда:

```
(isegeo:create-cuboid ((position 0 0 0) (position 10 20 5) "Aluminum" "Metall")
```

Созданные объекты можно модифицировать: срезать или скруглять углы, передвигать стороны и грани, и др. . Наиболее часто используются команды скругления и срезания углов.

Примеры:

```
(sdegeo:fillet-2d <угол> <радиус>)
```

```
(sdegeo:chamfer-2d <угол> <дистанция_среза>)
```

Переключение режимов «приоритет нового» / «приоритет старого»

```
(sdegeo:set-default-boolean "ABA" | "BAB" | "AB")
```

Если вновь созданный объект пересекается с ранее созданными, то он, в зависимости от установленного командой режима выполняет следующие операции: объединяется с ранее созданными – режим "AB"; замещает ранее

созданные объекты – режим "АВА"; уменьшается до пределов границ других объектов – режим "ВAB".

4.3.3. Задание профилей распределения примесей

Задание равномерного профиля

```
(sdedr:define-constant-profile "имя_профиля" "примесь" <значение>)
```

Для определения равномерного профиля распределения примеси требуется указать ее имя "имя_профиля", тип примеси "примесь" и концентрацию <значение>.

Локализация равномерного профиля

```
(sdedr:define-constant-profile-<тип> "имя_распределения" "имя_профиля"  
"имя_области" | "имя_материала")
```

Предыдущая команда лишь объявляла профиль, команда же `sdedr:define-constant-profile` определяет место его локализации. Если в качестве <тип> указать `region`, то профиль будет задан для указанной области "имя_области", если `material` – то для материала "имя_материала". Помимо этого, также требуется указать имя профиля "имя_профиля" и имя текущего распределения "имя_распределения", так как один и тот же профиль может быть присвоен несколько раз.

Примеры:

```
(sdedr:define-constant-profile "Epi-Prof-Subs" "BoronActiveConcentration" 1e15)
```

```
(sdedr:define-constant-profile-region "Place-Epi-Subs" "Epi-Prof-Subs" "Substrate")
```

Задание гауссовского профиля

```
(sdedr:define-gaussian-profile "имя_профиля" "примесь" "PeakPos" <значение>  
"PeakVal" <значение> "ValueAtDepth" <значение> "Depth" <значение>  
"Gauss" "Factor" <значение>)
```

В случае распределения Гаусса помимо "имя_профиля" и "примесь" также необходимо указать параметры гауссианы: глубину максимума распределения "PeakPos", концентрацию при максимуме "PeakVal", концентрацию на некоторой глубине "ValueAtDepth" и саму эту глубину "Depth". Ключевое слово "Gauss" определяет собственно гауссовский характер распреде-

ления, а значение для "Factor" показывает величину бокового ухода примеси относительно вертикального направления.

Локализация гауссовского профиля

```
(sdedr:define-refeval-window "имя_области" "тип_области"  
(position x1 y1 z1) (position x2 y2 z2) )  
(sdedr:define-analytical-profile-placement "имя_распределения"  
"имя_профиля" "имя_области" "Symm" "NoReplace" "Eval")
```

Для локализации гауссовского профиля используются сразу две команды. Первая определяет геометрическую область, на которую в дальнейшем будет наложен профиль примеси. "Тип_области" может быть прямоугольником Rectangle, линией Line или параллелепипедом Cuboid. Вне зависимости от типа области ее местоположение задается стандартными полями (position ...). Вторая команда проводит «увязку» профиля распределения с созданной областью. Помимо имен связываемых профиля "имя_профиля" и области "имя_области" также следует указать имя самого распределения "имя_распределения".

Примеры:

```
(sdedr:define-gaussian-profile "Gauss-N+" "ArsenicActiveConcentration"  
"PeakPos" 0.4 "PeakVal" 1e19 "ValueAtDepth" 1e15  
"Depth" 1.2 "Gauss" "Factor" 0.4)
```

```
(sdedr:define-refinement-window "Ref-N+_Source" "Line"  
(position 2.2 0 0) (position 2.5 0 0) )
```

```
(sdedr:define-analytical-profile-placement "Place-N+_Source" "Gauss-N+" "Ref-  
N+_Source" "Positive" "NoReplace" "Eval")
```

```
(sdedr:define-gaussian-profile "Gauss-P+" "BoronActiveConcentration"  
"PeakPos" 0 "PeakVal" P_CONC_SUBS "ValueAtDepth" CONC_EPI  
"Depth" DISTORT_LAYER "Gauss" "Factor" 0.8)
```

```
(sdedr:define-refeval-window "Ref-P+" "Rectangle"  
(position 0 D_EPI 0) (position X_SIZE Y_SIZE 0) )
```

```
(sdedr:define-analytical-profile-placement "Place-P+" "Gauss-P+" "Ref-P+"  
"Symm" "NoReplace" "Eval")
```

4.3.4. Определение контактов

Создание и определение контактов

```
(sdegeo:define-contact-set "имя")  
(sdegeo:set-current-contact-set "имя")
```

```
(sdegeo:define-2d-contact (find-edge-id (position x y z))  
(sdegeo:get-current-contact-set))
```

Процедура создания контактов в модуле **SDE** использует сразу три команды. Первая команда собственно создает контакт. Вторая команда определяет контакт как «текущий». Третья устанавливает местоположения «текущего» контакта. В случае, если контактом нужно назначить определенную область, последняя команда заменяется:

```
(sdegeo:set-contact-boundary-edges (find-region-id "имя_области"))
```

Примеры:

```
(sdegeo:define-contact-set "Drain")
```

```
(sdegeo:set-current-contact-set "Drain")
```

```
(sdegeo:define-2d-contact (find-edge-id (position 12 4 0))
```

```
(sdegeo:get-current-contact-set))
```

```
(sdegeo:define-contact-set "Gate")
```

```
(sdegeo:set-current-contact-set "Gate")
```

```
(sdegeo:set-contact-boundary-edges (find-region-id "PolyGate"))
```

Сохранение структуры

```
(sdeio:save-tdr-bnd (get-body-list) "@tdrboundary/o@")
```

Сформированные элементы конструкции сохраняются в так называемый граничный файл, содержащий описание координат границ всех конструктивных элементов модели. В случае если модуль **SDE** используется в дополнение к другому модулю моделирования, например, **Sprocess**, и часть контактов была определена там, то данные по этим контактам также будут добавлены в сохраняемый файл.

4.3.5. Построение расчетной сетки

Создание и задание параметров области определения сетки

```
(sdedr:define-refeval-window "имя_области" "тип_области"  
(position x1 y1 z1) (position x2 y2 z2) )
```

```
(sdedr:define-refinement-size "имя_размеров" {размеры_элемента})
```

```
(sdedr:define-refinement-function "имя_размеров" "поле"  
"функция_оптимизации" <значение>)
```

```
(sdedr:define-refinement-placement "имя_размещения" "имя_размеров"  
"имя_области")
```

SDE может запустить редактор сетки **SNMesh** как процедуру, таким образом делая необязательным его включение в проект. Несмотря на различия синтаксиса основные параметры и идеология **SDE**-модуля идентичны **SNMesh**. Более полно вопрос построения сетки будет рассмотрен в главе 5.

Пример:

```
(sdedr:define-refeval-window "Ref-Drain" "Rectangle"  
(position 5 0 0) (position 6.5 0.3 0) )
```

```
(sdedr:define-refinement-size "Size- Drain " 0.05 0.05 0.008 0.008)
```

```
(sdedr:define-refinement-function "Size-Drain" "DopingConcentration"  
"MaxTransDiff" 1)
```

```
(sdedr:define-refinement-placement "Ref-Place- Drain" "Size- Drain" "Ref-  
Drain")
```

*Вызов модуля **SNMesh** и сохранение сетки*

```
(sde:build-mesh "snmesh" " " "n@node@")
```

Данная команда осуществляет вызов модуля **SNMesh** и сохраняет построенную сетку в файл со стандартным именем `n@node@` .

4.4. Пример командного файла модуля SDE

Пример командного файла модуля **SDE** для моделирования MOS-транзистора, описанного в разделе 3.4, приведён в листинге 4.1.

Листинг 4.1

```
;----- Очистка проекта от предыдущих результатов-----  
(sde:clear)
```

```
;-----Определение внутренних параметров-----
```

```
(define L_N_PLUS      @L_N_plus@)  
(define L_GATE        @L_Gate@)  
  
(define C_SUB         1e15)  
(define C_WELL        7e17)  
  
(define Y_MAX         5)  
(define X_MAX         (+ L_GATE (* L_N_PLUS 2)))  
(define D_OX          -0.016)  
(define D_ME          -0.2)  
(define L_ME          0.3)
```

```
(define L_CON          0.2)
(define Y_BOT_GATE    (- D_OX 0.2))
(define X1_GATE       L_N_PLUS)
(define X2_GATE       (+ X1_GATE L_GATE))
```

-----Задание геометрии структуры-----

```
(sdegeo:set-default-boolean "ABA")
; кремниевая подложка
(sdegeo:create-rectangle (position 0 0 0) (position X_MAX Y_MAX 0)
"Silicon" "Substrate")

; подзатворный диэлектрик
(sdegeo:create-rectangle (position L_CON 0 0) (position (- X_MAX
L_CON) D_OX 0) "SiO2" "OX1")

; переключение в режим «приоритет старого»
(sdegeo:set-default-boolean "BAB")
; поликремниевый затвор
(sdegeo:create-rectangle (position X1_GATE 0 0) (position X2_GATE
Y_BOT_GATE 0) "PolySi" "PolyGate")

; окисел на затворе
(sdegeo:create-rectangle (position (- X1_GATE 0.02) 0 0)
(position (+ X2_GATE 0.02) (- Y_BOT_GATE 0.02) 0) "SiO2" "OX2")

; металлические контакты к истоку/стоку
(sdegeo:create-rectangle (position 0 0 0) (position L_ME D_ME 0)
"Aluminum" "Source")
(sdegeo:create-rectangle (position (- X_MAX L_ME) 0 0)
(position X_MAX D_ME 0) "Aluminum" "Drain")

; скругление углов затворного окисла
(sdegeo:fillet-2d (find-vertex-id (position (- X1_GATE 0.02)
(- Y_BOT_GATE 0.02) 0)) 0.02)
(sdegeo:fillet-2d (find-vertex-id (position (+ X2_GATE 0.02)
(- Y_BOT_GATE 0.02) 0)) 0.02)
```

-----Задание контактов-----

```
; контакт к затвору
(sdegeo:define-contact-set "Gate")
(sdegeo:set-current-contact-set "Gate")
(sdegeo:set-contact-boundary-edges (find-region-id "PolyGate"))

; контакт к истоку
```

```

(sdegeo:define-contact-set "Source")
(sdegeo:set-current-contact-set "Source")
(sdegeo:define-2d-contact (find-edge-id (position L_ME D_ME 0))
(sdegeo:get-current-contact-set))

; контакт к стоку
(sdegeo:define-contact-set "Drain")
(sdegeo:set-current-contact-set "Drain")
(sdegeo:define-2d-contact (find-edge-id (position X_MAX D_ME 0))
(sdegeo:get-current-contact-set))

; сохранение контактов
(sdeio:save-tdr-bnd (get-body-list) "@tdrboundary/o@")

;-----Определение концентрационных профилей-----

; определение равномерного профиля подложки
(sdedr:define-constant-profile "Prof-Subs" "PhosphorusActiveConcentration"
C_SUB)
; присвоение профиля подложке
(sdedr:define-constant-profile-region "Place-Subs" "Prof-Subs" "Substrate")

; определение профиля поликремниевого затвора
(sdedr:define-constant-profile "Prof-Poly" "PhosphorusActiveConcentration"
1e20)
(sdedr:define-constant-profile-region "Place-Poly" "Prof-Poly" "PolyGate")

; определение профиля р-кармана
(sdedr:define-gaussian-profile "Prof-P-well" "BoronActiveConcentration"
"PeakPos" 0.06 "PeakVal" C_WELL "ValueAtDepth" C_SUB "Depth" 1.04
"Gauss" "Factor" 0.8)

; определение места положения профиля р-кармана
(sdedr:define-refeval-window "Ref-Prof-P-well" "Line" (position 0 0 0)
(position X_MAX 0 0) )

; присвоение профиля р-карману
(sdedr:define-analytical-profile-placement "Place-P-well" "Prof-P-well"
"Ref-Prof-P-well" "Symm" "NoReplace" "Eval")

; определение профиля LDD-области
(sdedr:define-gaussian-profile "Prof-LDD" "ArsenicActiveConcentration"
"PeakPos" 0 "PeakVal" 4e19 "ValueAtDepth" 2e17 "Depth" 0.05
"Gauss" "Factor" 0.8)

```

```

; определение места положения профиля LDD-области слева от затвора
(sdedr:define-refeval-window "Ref-Prof-LDD1" "Line" (position 0 0 0)
(position X1_GATE 0 0) )

; присвоение профиля LDD-области слева от затвора
(sdedr:define-analytical-profile-placement "Place-LDD1" "Prof-LDD"
"Ref-Prof-LDD1" "Symm" "NoReplace" "Eval")

; определение места положения профиля LDD-области справа от затвора
(sdedr:define-refeval-window "Ref-Prof-LDD2" "Line" (position X2_GATE 0 0)
(position X_MAX 0 0) )

; присвоение профиля LDD-области справа от затвора
(sdedr:define-analytical-profile-placement "Place-LDD2" "Prof-LDD"
"Ref-Prof-LDD2" "Symm" "NoReplace" "Eval")

; определение профиля p+-областей
(sdedr:define-gaussian-profile "Prof-N_plus" "ArsenicActiveConcentration"
"PeakPos" 0 "PeakVal" 3e20 "ValueAtDepth" 2e17 "Depth" 0.1
"Gauss" "Factor" 0.8)

; определение места положения профиля истока
(sdedr:define-refeval-window "Ref-Prof-Source" "Line" (position 0 0 0)
(position (- X1_GATE 0.08) 0 0) )

; присвоение профиля истока
(sdedr:define-analytical-profile-placement "Place-Source" "Prof-N_plus"
"Ref-Prof-Source" "Symm" "NoReplace" "Eval")

; определение места положения профиля стока
(sdedr:define-refeval-window "Ref-Prof-Drain" "Line" (position (+ X2_GATE
0.08)
0 0) (position X_MAX 0 0) )

; присвоение профиля стока
(sdedr:define-analytical-profile-placement "Place-Drain" "Prof-N_plus"
"Ref-Prof-Drain" "Symm" "NoReplace" "Eval")

;-----Настройка сетки-----

; окисел и поликремний
(sdedr:define-refeval-window "Ref-Ox" "Rectangle" (position 0 0 0) (position
X_MAX -1 0) )
(sdedr:define-refinement-size "Size-Ox0.2 0. 0.02 0.02)
(sdedr:define-refinement-function "Size-Ox" "DopingConcentration"

```

```
"MaxTransDiff" 1)
(sdedr:define-refinement-placement "Ref-Place-Ox" "Size-Ox" "Ref-Ox" )
```

; кремниевая подложка

```
(sdedr:define-refinement-size "Size-Subs" 0.3 0.5 0.05 0.08)
(sdedr:define-refinement-function "Size-Subs" "DopingConcentration"
"MaxTransDiff" 1)
(sdedr:define-refinement-region "Ref-Place-Subs" "Size-Subs" "Substrate" )
```

; p-карман

```
(sdedr:define-refeal-window "Ref-well" "Rectangle" (position 0 0 0) (position
X_MAX 1.5 0) )
(sdedr:define-refinement-size "Size-well" 0.1 0.2 0.02 0.04)
(sdedr:define-refinement-function "Size-well" "DopingConcentration"
"MaxTransDiff" 1)
(sdedr:define-refinement-placement "Ref-Place-well" "Size-well" "Ref-well" )
```

; подзатворная (канальная) область

```
(sdedr:define-refeal-window "Ref-Channel" "Rectangle" (position X1_GATE 0
0)
(position X2_GATE 0.05 0) )
(sdedr:define-refinement-size "Size-Channel" 0.01 0.01 0.001 0.001)
(sdedr:define-refinement-function "Size-Channel" "DopingConcentration"
"MaxTransDiff" 1)
(sdedr:define-refinement-placement "Ref-Place-Channel" "Size-Channel"
"Ref-Channel" )
```

; n⁺-области сток/исток

```
(sdedr:define-refeal-window "Ref-Source" "Rectangle" (position 0 0 0) (posi-
tion X1_GATE 0.2 0) )
(sdedr:define-refeal-window "Ref-Drain" "Rectangle" (position X2_GATE 0 0)
(position X_MAX 0.2 0) )
(sdedr:define-refinement-size "Size-N_plus" 0.02 0.04 0.004 0.008)
(sdedr:define-refinement-function "Size-N_plus" "DopingConcentration"
"MaxTransDiff" 1)
(sdedr:define-refinement-placement "Ref-Place-Source" "Size-N_plus"
"Ref-Source" )
(sdedr:define-refinement-placement "Ref-Place-Drain" "Size-N_plus"
"Ref-Drain" )
```

; Создание и сохранение сетки

```
(sde:build-mesh "snmesh" " " "n@node@")
```

В результате выполнения командного файла 4.1 модуля SDE получается структура *n*-MOS-транзистора, приведённая на рисунке 4.2.

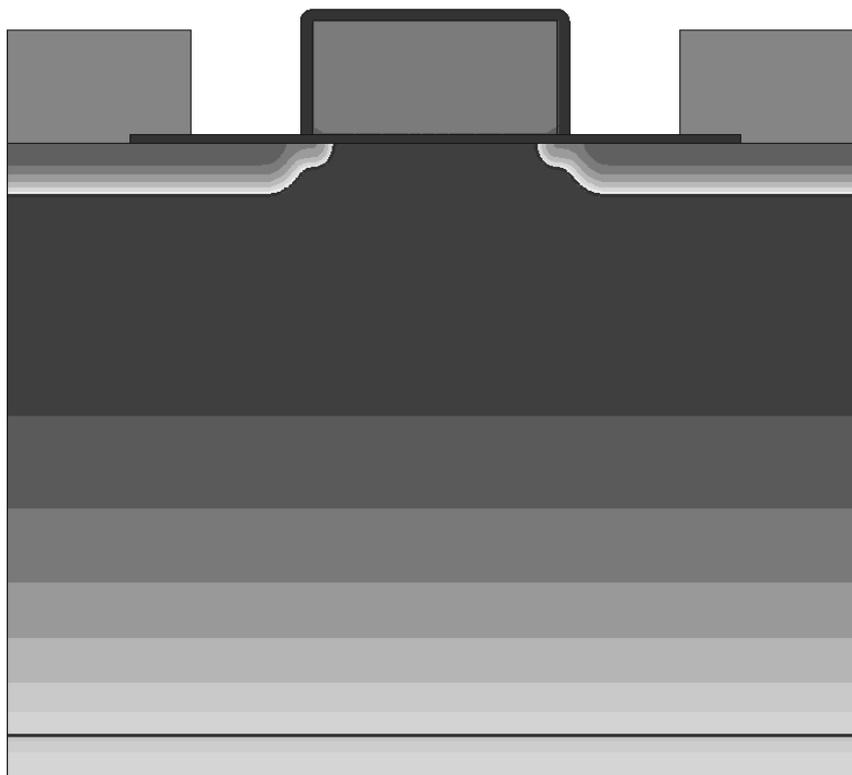


Рис. 4.2. Рассчитанная в модуле SDE структура n-MOS-транзистора

5. ОПТИМИЗАТОР РАСЧЕТНОЙ СЕТКИ SNMESH

5.1. Назначение программного модуля SNMesh

SProcess создает сетку, оптимизированную для расчета технологии изготовления прибора. Для того чтобы моделировать электро-, теплофизические и другие параметры и характеристики рассчитанной в **SProcess** структуры необходимо изменить расчетную сетку **Sprocess**. Для уменьшения времени последующих расчетов и с целью обеспечения точности решения фундаментальной системы уравнений в модуле **SDevice** нужно значительно увеличить размеры ячеек сетки в неактивных областях, например, в области равномерно легированной подложки. В то же время, в активных областях и особенно вблизи $p-n$ -переходов необходима значительно более мелкая сетка для более точного расчета параметров, но не чрезмерно мелкая, иначе системы уравнений перестают сходиться. Другими словами, сетка должна быть оптимизирована с учетом особенностей профиля распределения примеси.

Модуль **SNMesh** включает в себя несколько частей: редактор границ; редактор структур; редактор распределения примесей; оптимизатор сетки, который является одной из функций редактора распределения примесей. **SNMesh** позволяет создать одно-, двух- и трехмерные структуры, т.е. фактически нарисовать структуру «с нуля», не выполняя моделирование технологии изготовления. Созданная таким образом структура имеет простейшие профили распределения примесей, например, гауссианы, а также другие упрощенные технологические характеристики. Также ее можно применять для исследования каких-либо общих закономерностей как идеализированную модель.

5.2. Запуск и использование модуля SNMesh

Запуск модуля **SNMesh** удобнее производить в составе проекта, созданного в программе-оболочка **Workbench**. При этом необходимо обратить внимание на вкладку *Properties* контекстного меню, щелчок по которой вызывает окно *Tool properties for snmesh* (рис. 5.1). Во вкладке *Input Files* необходимо установить использование командного файла, созданного пользователем: *Master File*.

Чтобы запустить **SNMesh** как редактор границ необходимо выбрать файл типа *boundary* и щелкнуть по кнопке *Edit*.

Графическое окно модуля **SNMesh** показано на рисунке 5.2. В левом нижнем углу окна находится переключатель режимов работы **Boundary** – **Doping** (редактор структур – редактор распределения примесей).

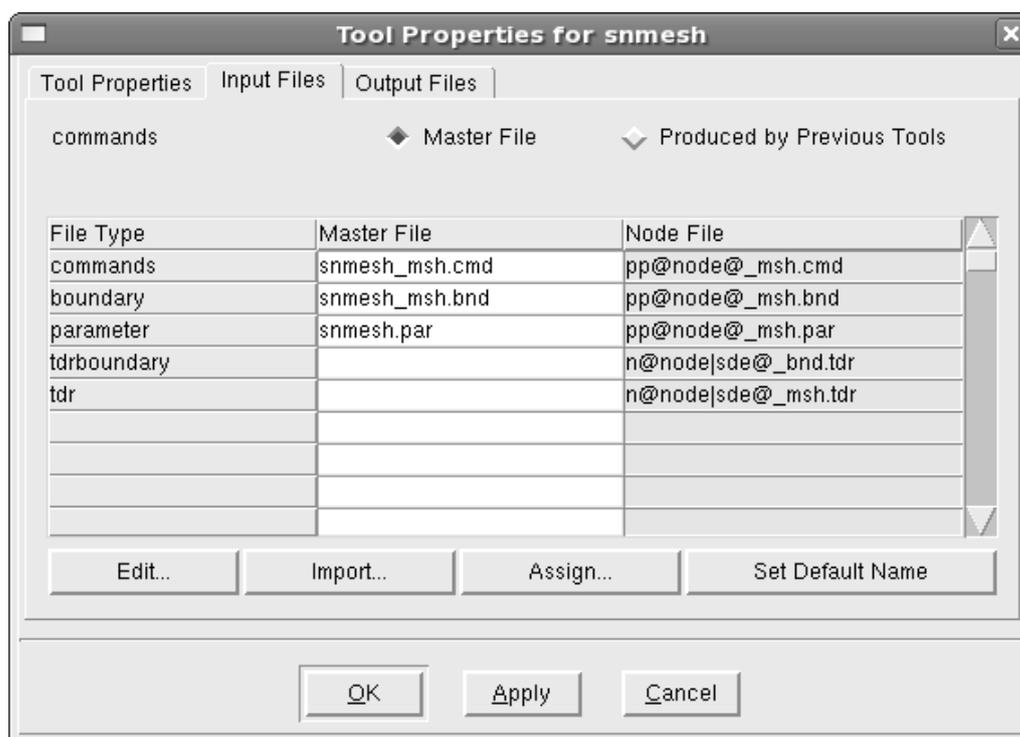


Рис. 5.1. Окно настройки параметров запуска программы **SNMesh**

Графическое окно редактора границ показано на рисунке 5.2. Редактор представляет собой модуль **SENTAURUS Structure Editor** и работа в нем проводится так же, как и в **SDE** в интерактивном режиме (рис 4.1).

Редактор границ поддерживает все возможности **SDE**: определение границ различных областей и материалов; профилей распределения примеси; положения контактов. Практически все эти функции доступны через панель инструментов, разбитую на три части: под строкой меню, а также слева и справа от рабочего поля. Кнопки управления (рис. 5.2, поз. 1) позволяют создавать объемные объектов: прямоугольник, цилиндр, шар и др. Для создания двумерных фигур служат кнопки позиции 2. Элементы на позиции 3 (рис. 5.2) позволяют изменять режим «приоритет нового» / «приоритет старого» (как команда `sdegeo:set-default-boolean`). Для удобства определения геометрических размеров создаваемых областей можно изменять положение системы координат относительно «наблюдателя» (рис. 5.2, поз. 4). При помощи элементов управления (рис. 5.2, поз. 5) создаются

области определения (команда `sdedr:define-refeval-window`), для которой может быть задан профиль распределения примеси при помощи кнопок управления (рис. 5.2, поз. 6).

Создание сетки выполняется при помощи команды меню **Mesh -> Build Mesh**.

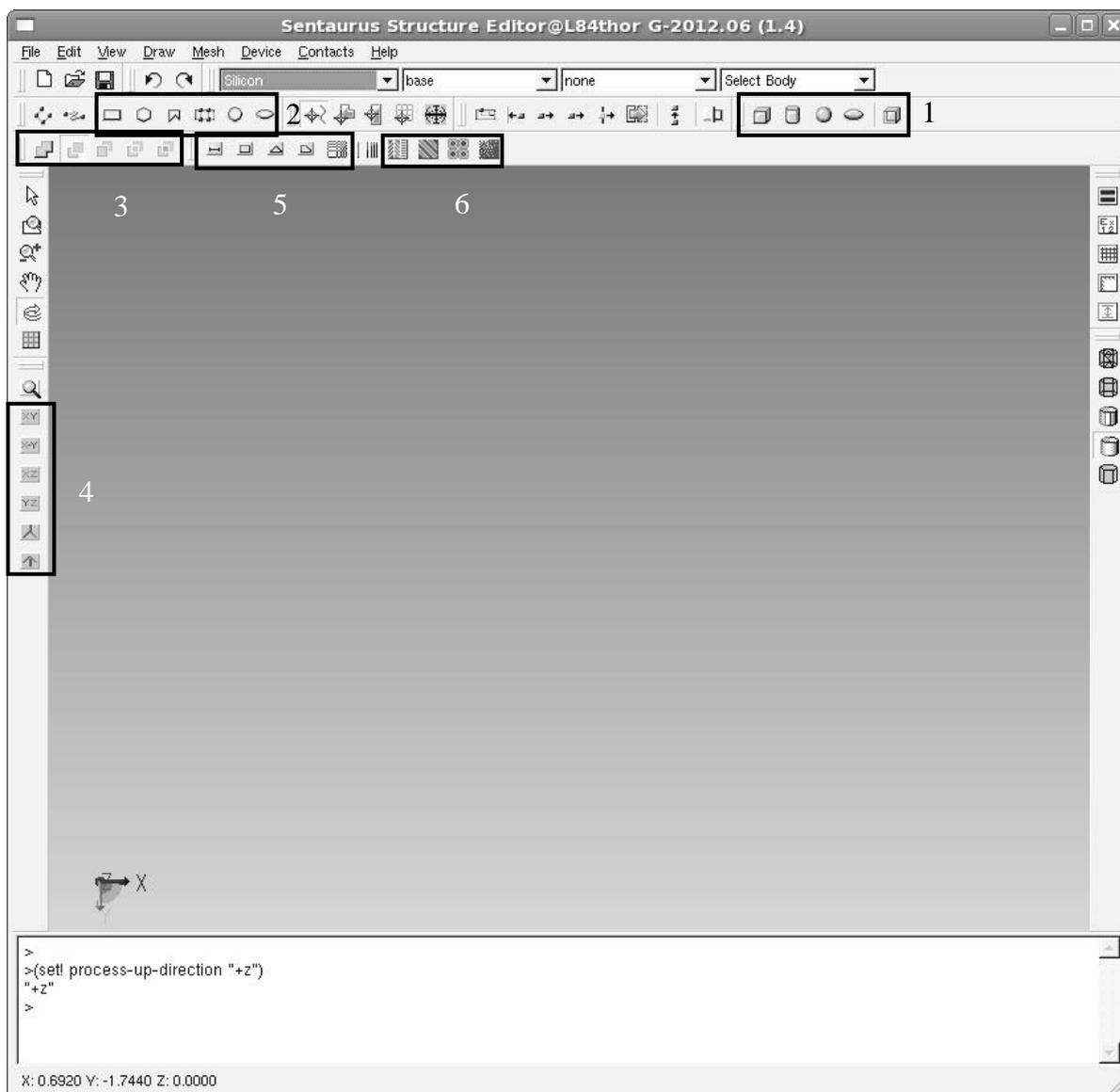


Рис. 5.2. Графическое окно редактора структур:

- 1 – создание объемных объектов; 2 – создание двухмерных фигур;*
- 3 – переключение режима «приоритет нового»/«приоритет старого»;*
- 4 – изменение положения системы координат; 5 – задание области определения; 6 – задание профиля распределения примеси*

Работа в графическом режиме специфична и плохо поддается автоматизации расчетов. Поэтому в большинстве случаев намного удобней

использовать командные файлы для соответствующих программных модулей: для создания структуры модуль **SDE**; для оптимизации сетки модуль **SNMesh**.

5.3. Структура командного файла модуля SNMesh

Командный файл модуля **SNMesh**, используемый для оптимизации расчетной сетки двумерной структуры, имеет следующую структуру:

```
Title "Заголовок"
#Описание областей, в которых будет оптимизироваться сетка
#с заданием основных характеристик сетки:
Definitions {
  Refinement "Default Region" #сетка для всей структуры (по умолчанию)
  { MaxElementSize = (Xmax1 Ymax1) #максимальный размер ячейки
    MinElementSize = (Xmin1 Ymin1) #минимальный размер ячейки
    RefineFunction = MaxTransDiff(Variable = "DopingConcentration", Value = 1.0)
      #функция оптимизации (в данном случае под профиль
      #распределения концентрации примесей) }
  Refinement "Region 1" #область с другим размером ячейки сетки
  { MaxElementSize = (Xmax2 Ymax2)
    MinElementSize = (Xmin2 Ymin2)
    RefineFunction = MaxTransDiff(Variable = "DopingConcentration", Value = 1.0) }
# .
# . аналогичное описание остальных областей
# .
#Определение файлов структуры, для которой будет оптимизироваться сетка:
  SubMesh "Structure"
  { Geofile = "@tdr@" #файл сетки Sprocess }
#Задание геометрического расположения объявленных ранее областей:
Placements {
  Refinement "Default Region" #вся структура
  { Reference = "Default Region"}
  Refinement "Region 1" #область с другим размером ячеек
  { Reference = "Region 1"
    RefineWindow = rectangle [(Xleft Ytop) , (Xright Ybottom)]
#определение прямоугольной области путем задания координат
# верхнего левого и нижнего правого углов }
# .
# . описание расположения остальных областей
# .
#Структура, относительно которой задано расположение областей:
  Submesh "Structure"
  { Reference = "Structure" } }
```

Такой командный файл подходит для оптимизации расчетной сетки большинства двумерных структур. Необходимо только устанавливать ми-

нимальные и максимальные размеры ячеек, добавлять необходимое количество областей с шагом, отличным от заданного по умолчанию для всей структуры, а также задавать геометрическое положение этих областей. У каждой области должно быть свое уникальное имя. При определении координат прямоугольной области необходимо учитывать, что в модуле **SNMesh** используется система координат с осью Y , направленной вниз, а не вверх, как в модуле **SProcess**.

5.4. Пример командного файла модуля SNMesh

Для оптимизации расчетной сетки структуры n -MOS-транзистора, полученной в результате выполнения командного файла 4.1 модуля **SDE**, написан командный файл (листинг 5.1) модуля **SNMesh**.

Листинг 5.1

```
Title "MOS"

# Описание размеров сетки для различных областей структуры
Definitions {

# Загрузка структуры
  SubMesh "SubMesh_0"
  { Geofile = "@tdr@" }
  # Кремниевая подложка
  Refinement "Substrate"
  { MaxElementSize = (0.7 0.4)   MinElementSize = (0.1 0.08)
    RefineFunction = MaxTransDiff(Variable = "DopingConcentration", Value = 1) }

# P-карман
  Refinement "P-well"
  { MaxElementSize = (0.3 0.2)   MinElementSize = (0.02 0.01)
    RefineFunction = MaxTransDiff(Variable = "DopingConcentration", Value = 1) }

# N+- области
  Refinement "N_plus"
  { MaxElementSize = (0.02 0.02)   MinElementSize = (0.003 0.003)
    RefineFunction = MaxTransDiff(Variable = "DopingConcentration", Value = 0.3) }

# Канальная область
  Refinement "Channel"
  { MaxElementSize = (0.005 0.005)   MinElementSize = (0.001 0.001)
    RefineFunction = MaxTransDiff(Variable = "DopingConcentration", Value = 1) }

# Поликремниевый затвор
  Refinement "PolySilicon"
  { MaxElementSize = (0.1 0.1)   MinElementSize = (0.01 0.01)
    #RefineFunction = MaxTransDiff(Variable = "DopingConcentration", Value = 1) }

# Окисел
  Refinement "Oxide"
  { MaxElementSize = (0.1 0.1)   MinElementSize = (0.01 0.01)
```

```

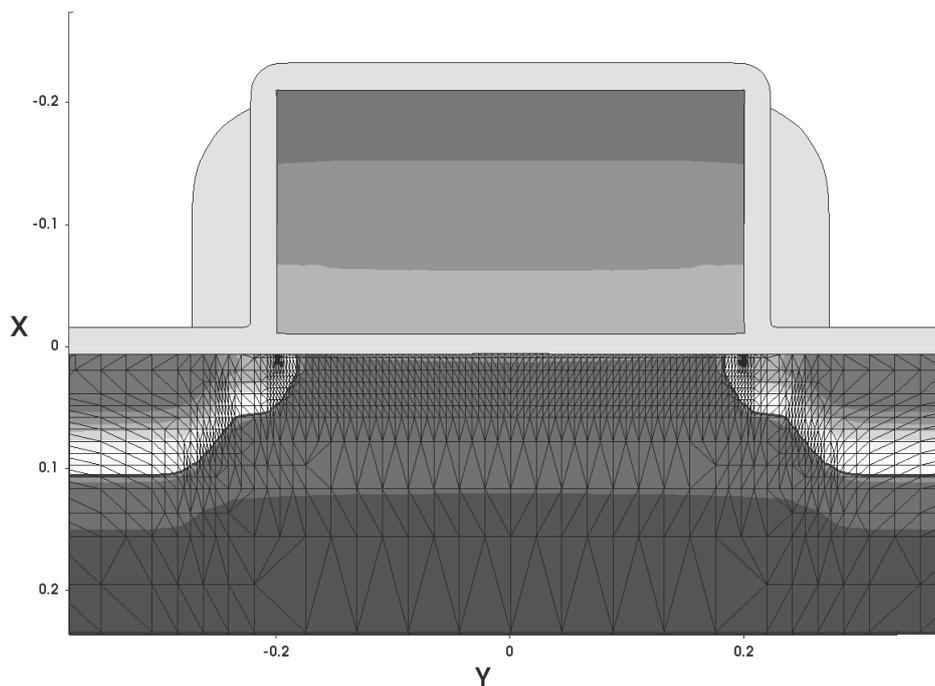
    #RefineFunction = MaxTransDiff(Variable = "DopingConcentration", Value = 1) }
# Нитридные спейсеры
  Refinement "Nitride"
  { MaxElementSize = (0.1 0.1) MinElementSize = (0.01 0.01)
    #RefineFunction = MaxTransDiff(Variable = "DopingConcentration", Value = 1) } }
# Описание месторасположения областей
Placements {
# Вся структура
  Submesh "SubMesh_0"
  { Reference = "SubMesh_0" }
# Кремниевая подложка
  Refinement "Substrate"
  { Reference = "Substrate" }
# P-карман
  Refinement "P-well"
  { Reference = "P-well"
    RefineWindow = rectangle [( 0 @<-1*(L_Gate/2+L_N_plus)>@ ),
      ( 1.2 @<L_Gate/2+L_N_plus>@ ) ] }

# Истоковая n+ область
  Refinement "Source"
  { Reference = "N_plus"
    RefineWindow = rectangle [( 0 @<-1*(L_Gate/2+L_N_plus)>@ ),
      ( 0.2 @<-1*(L_Gate/2)>@ ) ] }
# Стоковая n+ область
  Refinement "Drain"
  { Reference = "N_plus"
    RefineWindow = rectangle [( 0 @<L_Gate/2>@ ),
      ( 0.2 @<L_Gate/2+L_N_plus>@ ) ] }
# Канальная область
  Refinement "Channel"
  { Reference = "Channel"
    RefineWindow = rectangle [( 0 @<-1*(L_Gate/2)>@ ),
      ( 0.05 @<L_Gate/2>@ ) ] }
# Поликремниевый затвор
  Refinement "PolySilicon"
  { Reference = "PolySilicon"
    RefineWindow = material ["PolySilicon"] }
# Окисел
  Refinement "Oxide"
  { Reference = "Oxide"
    RefineWindow = material ["Oxide"] }
# Нитридные спейсеры
  Refinement "Nitride"
  { Reference = "Nitride"
    RefineWindow = material ["Nitride"] } }

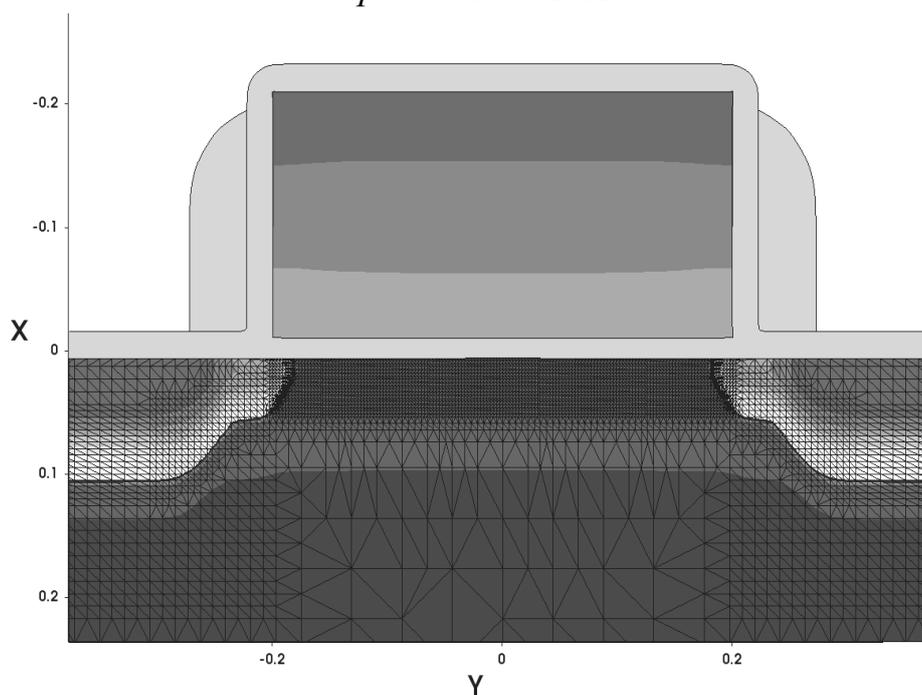
```

Результат выполнения командного файла можно просмотреть, вызвав контекстное меню: *Visualize* → *Sentaurus Visual (All Files)*. Структура с уже оптимизированной сеткой помещается в 2 файла: *_bnd.tdr и *_msh.tdr.

На рисунке 5.3 показана расчетная сетка структуры *n*-MOS-транзистора до оптимизации, на рисунке 5.4 – после оптимизации.



*Рис. 5.3. Расчетная сетка структуры *n*-MOS-транзистора до обработки в *SNMesh**



*Рис. 5.4. Структура *n*-MOS-транзистора с оптимизированной расчетной сеткой в модуле *SNMesh**

6. ПРОГРАММНЫЙ МОДУЛЬ SENTAURUS DEVICE ДЛЯ МОДЕЛИРОВАНИЯ ПАРАМЕТРОВ ПОЛУПРОВОДНИКОВЫХ ПРИБОРОВ

6.1. Назначение модуля SDevice

Программный модуль **SENTAURUS Device (SDevice)** предназначен для одно-, двух- и трехмерного моделирования полупроводниковых приборов с расчетом их электрических, температурных, оптических параметров и характеристик в различных условиях функционирования.

SDevice объединяет в себе современные физические модели и численные методы для моделирования большинства существующих на сегодняшний день полупроводниковых структур, начиная с субмикронных MOS-транзисторов и заканчивая гетероструктурами и лазерами. Модуль **SDevice** содержит множество моделей процессов, происходящих в полупроводниковых приборах: дрейфовая диффузия, термодинамические эффекты, гидродинамические модели и многие другие. **SDevice** также включает в себя большой набор средств для решения систем нелинейных уравнений, поддерживает Spice-модели.

SDevice моделирует работу полупроводниковых приборов, основываясь на уравнениях физики полупроводников, описывающих механизмы проводимости и распределения носителей.

Базовой является диффузионно-дрейфовая модель

$$\left\{ \begin{array}{l} \nabla(\varepsilon \cdot \nabla \psi) = -q(p - n + N_D^+ - N_A^-) \quad - \text{уравнение Пуассона}; \\ \nabla \vec{J}_n = q(R - G) + q \frac{\partial n}{\partial t} \\ -\nabla \vec{J}_p = q(R - G) + q \frac{\partial p}{\partial t} \end{array} \right\} \quad - \text{уравнения непрерывности}; \quad (6.1)$$

$$\left\{ \begin{array}{l} \vec{J}_n = -q(\mu_n n \cdot \nabla \psi - D_n \nabla n) \\ \vec{J}_p = -q(\mu_p p \cdot \nabla \psi + D_p \nabla p) \end{array} \right\} \quad - \text{уравнения для электронного и дырочного токов},$$

где ε – диэлектрическая проницаемость; ψ – электростатический потенциал; q – заряд электрона; p – концентрация дырок; n – концентрация электронов; N_D^+ – концентрация ионизированных атомов доноров; N_A^- – концентрация ионизированных атомов акцепторов; J_n – плотность электронного тока; J_p – плотность дырочного тока; R – скорость рекомбинации электронов и дырок; G – скорость генерации электронов и дырок; μ_n, μ_p – подвижности

электронов и дырок соответственно; D_n, D_p – коэффициенты диффузии электронов и дырок соответственно.

Уравнения для электронного и дырочного токов в термодинамической модели учитывают термоэлектрические эффекты, связанные с неоднородным распределением температуры:

$$\begin{cases} \vec{J}_n = -q(\mu_n n \cdot \nabla \varphi_n - P_n \nabla T) \\ \vec{J}_p = -q(\mu_p p \cdot \nabla \varphi_p + P_p \nabla T) \end{cases}, \quad (6.2)$$

где φ_n, φ_p – квазиуровни Ферми для электронов и дырок соответственно; P_n, P_p – величины термоэдс для электронов и дырок соответственно; T – температура.

Уравнения для электронного и дырочного токов в гидродинамической модели с учетом уравнений переноса энергии для электронов и дырок составляют следующую систему уравнений:

$$\left. \begin{cases} \vec{J}_n = -q(\mu_n n \cdot \nabla E_c - k_B T_n \nabla n) \\ \vec{J}_p = -q(\mu_p p \cdot \nabla E_v + k_B T_p \nabla p) \\ \frac{\partial W_n}{\partial t} + \text{div} S_n = J_n \nabla E_c + \frac{\partial W_n}{\partial t} \Big|_{CT} \\ \frac{\partial W_p}{\partial t} + \text{div} S_p = J_p \nabla E_v + \frac{\partial W_p}{\partial t} \Big|_{CT} \\ \frac{\partial W_l}{\partial t} + \text{div} S_l = \frac{\partial W_l}{\partial t} \Big|_{CT} \end{cases} \right\} \begin{array}{l} \text{уравнения переноса энергии} \\ \text{для электронов, дырки решётки,} \end{array} \quad (6.3)$$

где E_c, E_v – энергии зоны проводимости и валентной зоны соответственно; T_n, T_p, T_l – эффективная температура электронов, дырок и решётки соответственно; k_B – постоянная Больцмана; W_n, W_p, W_l и S_n, S_p, S_l – энергия и энергетический поток электронов, дырок и решётки соответственно.

Реальный полупроводниковый прибор представляется в **SDevice** как "виртуальный прибор", свойства которого дискретизированы по узлам неоднородной сетки. Следовательно, непрерывные функции, такие как профили легирования, разбиты на конечное число дискретных точек, и значения этих функций между узлами сетки могут быть получены путем интерполяции. Граничные и начальные условия задаются путем указания напряжений, токов, зарядов, температур и других параметров на электрических и тепловых контактах.

Для моделирования в **SDevice** используются структуры, созданные с помощью **SProcess**, **SNMesh** и других модулей. Структура должна быть с оптимизированной расчетной сеткой. **SDevice** не имеет графического интерфейса и управляется только командным файлом.

6.2. Структура командного файла модуля **SDevice**

Командный файл **SDevice** состоит из набора секций, порядок расположения которых обычно не имеет значения. **SDevice** не чувствителен к регистру. Строки, начинающиеся со знаков # и *, считаются комментариями.

Стандартный командный файл модуля **SDevice** имеет следующую структуру:

**Определение входных файлов моделируемой структуры и выходных файлов:*

```
File { *Задание входных файлов:
    grid = "@tdr@"
    *Задание выходных файлов:
    plot = "@tdrdat@"
    current = "@plot@"
    output = "@log@" }
```

**Задание электрических контактов и начальных условий для них:*

```
Electrode { {name = "electrode1" voltage = V1}
             {name = "electrode2" current = I1}
             .
             .
             . }
```

**Задание термических контактов и начальных условий для них:*

```
Termode { {name = "termode1" temperature = T1}
           .
           .
           . }
```

**Задание моделей физических процессов, используемых в расчете:*

```
Physics {
    .
    .
    . }
```

**Определение математических методов решения системы уравнений:*

```
Math {
    .
    .
    . }
```

**Описание параметров и характеристик, которые будут рассчитываться:*

```
Plot {  
  .  
  .  
  . }  
}
```

**Составление и решение необходимых систем уравнений:*

Solve { **Получение начального решения из начальных условий:*

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

**Решение системы уравнений при переменных параметрах*

**на электродах или термодах (или при других условиях):*

```
QuasiStationary( InitialStep = Step0
```

```
MaxStep = MaxStep MinStep = MinStep
```

```
Goal { name = "Наименование_контакта" parameter = Ре-  
зультатирующее_значение } )
```

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

5.3. Основные команды модуля SDevice

Основные параметры секций командного файла **SDevice** имеют следующие назначение и структуру:

File{...} – в данной секции определяются входные и выходные файлы **SDevice**. Входные файлы: файл сетки *.grd; файл распределения примесей *.dat моделируемой полупроводниковой структуры. Выходные файлы: файл *_des.dat, содержащий распределения различных параметров на входной сетке; файл *_des.plt, содержащий различные зависимости, представляемые в виде графиков; файл протокола *_des.log. При использовании **SDevice** в составе проекта в оболочке **Workbench** со стандартными именами файлов, например, n3_mdr.grd, n4_des.dat и т.д., удобно используемые в **SDevice** файлы обозначать следующим образом:

```
grid   = "@tdr@"  
plot   = "@tdrdat@"  
current = "@plot@"  
output = "@log@"
```

При таком определении будут автоматически подставлены файлы моделируемой структуры и созданы выходные файлы со стандартными именами. При использовании нестандартных имен необходимо явно указать имена файлов, например:

```
grid   = "nmos_mdr.grd"
```

```
doping = "nmos_mdr.dat"
plot   = "output_des.dat"
current = "output_des.plt"
output  = "output_des.log"
```

Файлы с расширением *_des.plt можно просматривать в программе **Inspect**, а файл *_des.dat вместе с соответствующим ему файлом *_mdr.grd - в программе **Tecplot SV**;

Electrode{...} – эта секция служит для определения электрических контактов моделируемой структуры и задания начальных параметров на них. Данные параметры служат электрическими граничными условиями при решении систем уравнений, описывающих электрофизические свойства моделируемого прибора. Названия электродов соответствуют названиям, данным в программе **SProcess** или **SNMesh**. Контакты, которые не описаны в данной секции, игнорируются **SDevice**. Следует также отметить, что хотя сам **SDevice** не чувствителен к регистру, при задании имен контактов это следует учитывать. В качестве начальных параметров на электродах могут быть заданы: напряжение Voltage, В, ток Current, А, заряд Charge, Кл и другие параметры. Можно также указать, что к электроду подключен резистор Resistor, Ом. При задании затвора MOS-транзистора можно указывать разность работ выхода между металлом затвора и кремнием barrier, эВ. При использовании затвора из сильнолегированного поликремния нужно указать тип проводимости поликремния material = "PolySi"(N).

Например:

```
Electrode { {name = "source" voltage = 0.0}
             {name = "gate"  voltage = -2.0 material = "PolySi"(N)}
             {name = "substrate" current = 1e-3}
             {name = "drain" voltage = 2.0 resistor = 100}
             {name = "gate1" voltage = 5.0 Barrier=-0.55} }
```

Termode{...} – секция для определения термических контактов. Если при расчете не учитываются термодинамические характеристики модели, то данная секция не используется. Секция по назначению и структуре аналогична секции **Electrode{ }**. В качестве начальных параметров чаще всего используются температура temperature, К, тепловое сопротивление SurfaceResistance, см²·К/Вт, коэффициент теплопроводности – величина, обратная тепловому сопротивлению SurfaceConductance и др.

Например:

```
Thermode { { Name = "substrate" Temperature = 300 }  
{name = "surface" Temperature = 310 SurfaceResistance = 0.1} } ;
```

Physics{...} – секция для описания используемых в расчете физических моделей. Основные параметры:

AreaFactor = *number* – задает толщину [мкм] двумерной структуры, преобразуя ее в трехмерную, по умолчанию 1 мкм;

Temperature = *number* – температура [K], при которой находится виртуальная структура, по умолчанию 300 K;

EffectiveIntrinsicDensity(BandGapNarrowing(Slotboom)) – модель сужения запрещенной зоны кремния, по умолчанию учет этого эффекта включен. Наиболее часто используется модель Slotboom, которую можно задавать в виде EffectiveIntrinsicDensity(Slotboom);

Hydrodynamic – включение гидродинамической транспортной модели;

Mobility(...) – задание различных моделей, учитывающих изменение подвижности носителей под действием различных факторов. Из них наиболее часто используются следующие:

DopingDependence(Masetti) – модели, учитывающие зависимость подвижности носителей от концентрации примесей, для кремния по умолчанию используется модель Masetti;

HighFieldSaturation(GradQuasiFermi) – модели насыщения дрейфовой скорости носителей в сильном электрическом поле, по умолчанию используется модель GradQuasiFermi, также часто используется модель Eparallel, учитывающая параллельное поверхности подложки электрическое поле;

NormalElectricField или Enormal - модель, учитывающая влияние нормального к поверхности подложки электрического поля;

CarrierCarrierScattering(ConwellWeisskopf) – модели, учитывающие рассеивание носителей на других носителях; по умолчанию используется модель ConwellWeisskopf;

Recombination(...) – используемые модели генерации-рекомбинации носителей, по умолчанию все модели отключены. Наиболее часто используются следующие модели:

SRH(*models*) – рекомбинация Шокли-Рида-Холла, *models* – учет зависимости от концентрации примесей *DopingDependence*, от температуры *TempDependence*, учет туннельного эффекта *Tunnelin* и др.;

CDL(*models*) – рекомбинация на парных дефектных энергетических уровнях, *models* – те же, что и в рекомбинации ШРХ;

Auger – рекомбинация Оже;

Band2Band – генерация носителей путем туннельного перехода зона–зона;

Avalanche(*VanOverstraeten GradQuasiFermi*) – лавинная генерация электронно-дырочных пар (или ударная ионизация); модель по умолчанию *VanOverstraeten*; модель напряженности внутреннего электрического поля *GradQuasiFermi* (по умолчанию) либо *Eparallel*;

SurfaceSRH – поверхностная рекомбинация ШРХ;

Thermodynamic – включение учета термодинамических характеристик моделируемой структуры, например, для учета саморазогрева;

Charge(Concentration = *number*) – концентрация зарядов в оксиде кремния, см^{-3} , по умолчанию 0 см^{-3} ;

Physics(MaterialInterface="Oxide/Silicon") {charge(surfconc=*Nss*)} – таким образом можно задать концентрацию зарядов на границе раздела оксид-кремний, см^{-3} . Подобным образом задаются и другие параметры и модели на границах раздела, для отдельных материалов, областей и т.д. Следует обратить внимание на то, что данные команды не включаются в основную секцию Physics{...}.

Для многих задач достаточно следующих моделей физических процессов:

```
Physics{ AreaFactor = number
        Temperature = number
        EffectiveIntrinsicDensity(Slotboom)
        Mobility( DopingDependence HighFieldSaturation Enormal )
        Recombination( SRH(DopingDependence) CDL(DopingDependence)
                    Band2Band Avalanche )
        Charge(Concentration = number) }
```

```
Physics(MaterialInterface="Oxide/Silicon") {charge(surfconc=Nss)} ;
```

Math{...} – секция, в которой описываются используемые для решения уравнений математические методы. Наиболее часто используемые параметры:

Cylindrical – указывает, что для моделируемой структуры необходимо использовать цилиндрическую систему координат. Это полезно при моделировании диодных и других простых двумерных структур;

Derivatives – использование аналитических производных подвижности носителей в решаемых уравнениях. По умолчанию этот параметр включен, для его выключения нужно указать Derivatives;

AvalDerivatives – использование аналитических производных в уравнениях ударной ионизации. По умолчанию этот параметр включен, для его выключения нужно указать AvalDerivatives;

NewDiscretization – использование улучшенной схемы дискретизации уравнений. По умолчанию этот параметр включен, для его выключения нужно указать NewDiscretization;

RelErrControl – включение относительного критерия остановки итерационного процесса при достижении необходимой точности, по умолчанию включен; задание RelErrControl включает абсолютный критерий, при этом необходимо задать параметр Digits=*number*. По умолчанию Digits = 5, означающий число знаков после запятой в критерии точности;

Iterations – задание максимального количества итераций на каждом шаге, по умолчанию Iterations=50. Если уравнения не сходятся после заданного количества итераций, то шаг уменьшается и процедура повторяется;

Extrapolate – использование экстраполяции при итерационном решении уравнений. По умолчанию параметр выключен. Рекомендуется его включать для улучшения сходимости уравнений;

Smooth – сглаживание результатов, полученных на соседних шагах, используется при плохой сходимости;

BreakCriteria – механизм прерывания расчета при выполнении определенного условия.

Наиболее простым примером может служить команда

```
BreakCriteria{Current(Contact="имя" AbsVal=значение)}
```

– прерывает расчет при возрастании абсолютного значения тока через указанный контакт до указанного уровня.

Для большинства случаев достаточно явно заданных параметров:

```
Math{ RelErrControl Iterations = 20 Extrapolate }
```

Plot{...} – определение физических величин, распределения которых будут рассчитаны и занесены в файл *_des.dat. Вычислены будут только те величины, расчет которых возможен на основе выбранных моделей физических процессов. Наиболее часто используемые величины:

AcceptorConcentration – концентрация акцепторных примесей;

AntimonyActiveConcentration-концентрация электрически активной сурьмы;

AntimonyConcentration – концентрация сурьмы;

AntimonyPlusConcentration – концентрация положительных ионов сурьмы;
ArsenicActiveConcentration – концентрация электрически активного мышьяка;
ArsenicConcentration – концентрация мышьяка;
ArsenicPlusConcentration – концентрация положительных ионов мышьяка;
AugerRecombination – скорость рекомбинации Оже;
AvalancheGeneration – скорость лавинной генерации носителей;
Band2BandGeneration – скорость генерации носителей путем туннельного перехода зона–зона;
BandGap – ширина запрещенной зоны;
BandgapNarrowing – сужение запрещенной зоны;
BoronActiveConcentration – концентрация электрически активного бора;
BoronConcentration – концентрация бора;
BoronMinusConcentration – концентрация отрицательных ионов бора;
BuiltinPotential – встроенный электрический потенциал;
CDLRecombination – скорость рекомбинации на парных уровнях дефектов;
ConductionBandEnergy – энергия на дне зоны проводимости;
DielectricConstant – диэлектрическая проницаемость;
DonorConcentration – концентрация доноров;
DopingConcentration – разность концентрации донорных и акцепторных примесей;
eCurrentDensity – плотность электронного тока;
eDensity – концентрация электронов;
eDirectTunnelCurrent – плотность туннельного электронного тока;
eDriftVelocity – дрейфовая скорость электронов;
EffectiveBandGap – эффективная ширина запрещенной зоны;
EffectiveIntrinsicDensity – эффективная собственная концентрация носителей;
eGradQuasiFermi – градиент электронного квазиуровня Ферми;
eQuasiFermiPotential – электронный квазипотенциал Ферми;
ElectricField – напряженность электрического поля;
ElectronAffinity – электронное сродство;
ElectrostaticPotential – электростатический потенциал;
eLifetime – время жизни электронов;
eMobility – подвижность электронов;
eSaturationVelocity – скорость электронов в области насыщения;
eTemperature – температура электронов;
eVelocity – скорость электронов;
hCurrentDensity – плотность дырочного тока;

hDensity – концентрация дырок;
 hDirectTunnelCurrent – плотность туннельного дырочного тока;
 hDriftVelocity – дрейфовая скорость дырок;
 hGradQuasiFermi – градиент дырочного квазиуровня Ферми;
 hLifetime – время жизни дырок;
 hMobility – подвижность дырок;
 HotElectronInj – инжекция горячих электронов;
 HotHoleInj – инжекция горячих дырок;
 hQuasiFermiPotential – дырочный квазипотенциал Ферми;
 hSaturationVelocity – скорость дырок в области насыщения;
 hTemperature – температура дырок;
 hVelocity – скорость дырок;
 LatticeTemperature – температура;
 PhosphorusActiveConcentration - концентрация электрически активного фосфора;
 PhosphorusConcentration – концентрация фосфора;
 PhosphorusPlusConcentration-концентрация положительных ионов фосфора;
 QuasiFermiPotential – квазипотенциал Ферми;
 SpaceCharge – пространственный заряд;
 SRHRecombination – скорость рекомбинации Шокли–Рида–Холла;
 ThermalConductivity – теплопроводность;
 ThomsonHeat – скорость выделения теплоты в результате эффекта Томсона;
 TotalConcentration – полная концентрация примесей;
 TotalCurrentDensity – полная плотность тока;
 TotalHeat – полная скорость выделения теплоты;
 TotalRecombination – полная скорость рекомбинации.

Пример:

```

Plot{ AcceptorConcentration DonorConcentration
      BoronConcentration PhosphorusConcentration TotalConcentration DopingCon-
      centration
      BandGap BandgapNarrowing BuiltInPotential
      ElectrostaticPotential SpaceCharge ElectricField
      SRHRecombination CDLRecombination AvalancheGeneration
      Band2BandGeneration TotalRecombination
      eDensity eMobility eCurrentDensity
      hDensity hMobility hCurrentDensity
      TotalCurrentDensity
      LatticeTemperature }
  
```

Solve{...} – секция для задания используемых уравнений и условий их решения. В данной секции важен порядок следования команд. Обычно используется следующая схема решения: получение начального решения, затем циклическое изменение с определенным шагом какого-либо начального параметра и расчет уравнений в режиме установившегося равновесия (квазистационарный режим). Для решения уравнений обычно используется алгоритм Ньютона. Начальное решение чаще всего формируется за один или два шага. Например, можно на первом шаге решить уравнение Пуассона с использованием начальных условий, заданных в секции Electrode{...}. На втором шаге решается система уравнений, состоящая из уравнения Пуассона и уравнений непрерывностей для электронов и дырок, в которой в качестве начальных условий используются результаты, полученные на первом шаге:

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

Такой подход обычно обеспечивает хорошую сходимость. Можно на первом же шаге решать систему уравнений, тогда отдельное решение уравнения Пуассона и второй шаг не нужны. При учете термодинамических свойств моделируемого прибора рекомендуется в начальное решение не включать уравнения термодинамики для обеспечения лучшей сходимости.

После расчета начального решения обычно используется следующая процедура: с определенным шагом на заданном электроде (термоде) изменяется начальный параметр, для каждого шага рассчитывается система уравнений coupled{Poisson Electron Hole} или coupled{Poisson Electron Hole Temperature} при учете термодинамических свойств в квазистационарных условиях, используя новое значение параметра вместо исходного. В качестве начального используется решение, полученное на предыдущем шаге. Если система уравнений не сходится, то шаг изменения параметра уменьшается и система рассчитывается заново. После успешного решения шаг увеличивается. Процедура повторяется, пока параметр на электроде (термоде) не достигнет заданного значения. В случае систематической расходимости уравнений расчет прерывается. Параметры изменения шага: начальный InitialStep, минимальный MinStep и максимальный MaxStep шаги, а также коэффициенты увеличения Increment и уменьшения Decrement шага.

Например:

```
QuasiStationary ( InitialStep = 1e-3 MaxStep = 1e-2 MinStep = 1e-7
Goal { name = "gate" voltage = 5 })
{ Coupled {Hole Electron Poisson} }
```

В данном примере напряжение на электроде *gate* изменяется от начального до 5 В, шаг изменения и пределы его вариаций заданы параметрами *InitialStep*, *MaxStep* и *MinStep*. *Increment* и *Decrement* установлены по умолчанию равными 2. Параметры на остальных электродах остаются неизменными. На каждом шаге решается система, состоящая из уравнений Пуассона, непрерывности для электронов и дырок.

Примеры команды *Solve{...}*:

а) без учета термодинамических характеристик:

```
Solve{ Poisson
      Coupled {Poisson Electron Hole}
      QuasiStationary( InitialStep = 0.001 MaxStep = 0.01  MinStep = 1e-7
                      Goal { name = "gate" voltage = 5 })
      { Coupled {Hole Electron Poisson} } }
```

б) с учетом термодинамических характеристик (например, при учете само-разогрева):

```
Solve { Coupled {Poisson Electron Hole}
      QuasiStationary ( InitialStep = 0.01 MaxStep = 0.1  MinStep = 1e-5
                      Goal { name = "drain" voltage = 30 })
      { Coupled {Hole Electron Poisson Temperature} } }
```

В команде *Solve{...}* может быть несколько команд *Goal{...}*. После выполнения команды *Solve{...}* на использованном в команде *Goal{...}* электроде (термоде) остается то значение параметра, которое было последним в итерационной процедуре. В выходной файл **_des.dat* записываются указанные в секции *Plot{...}* величины, рассчитанные при этом последнем значении параметра на электроде. В файл **_des.plt* записываются данные по мере изменения значения рассматриваемого параметра. Поэтому различные распределения из файла **_des.dat* в **Tecplot SV** можно просмотреть только при фиксированном режиме моделируемого прибора. Характеристики из файла **_des.plt* можно просмотреть в **Inspect** во всем диапазоне изменения рассматриваемого параметра – это различные вольт-амперные и другие характеристики.

6.4. Пример командного файла модуля SDevice

Ниже приведен пример командного файла модуля **SDevice** (листинг 6.1) для расчета передаточной вольт-амперной характеристики *n*-MOS-транзистора.

Листинг 6.1

```
File { grid = "@grid@"
      doping = "@doping@"
      plot = "@dat@"
      current = "@plot@"
      output = "@log@" }
Electrode { {name = "source" voltage = 0.0}
           {name = "gate" voltage = -1.0 material = "PolySi"(N)}
           {name = "substrate" voltage = 0.0}
           {name = "drain" voltage = 10.0} }
Physics { AreaFactor = 1e3
          EffectiveIntrinsicDensity(Slotboom)
          Mobility( DopingDependence HighFieldSaturation(GradQuasiFermi)
                  NormalElectricField )
          Recombination( SRH(DopingDependence ) Band2Band )
          Temperature = 300 }
Physics( MaterialInterface="Oxide/Silicon" ) {charge(surfconc=1e11)}
Math { Extrapolate Iterations = 7 RelerrControl Derivatives NewDiscretization }
Plot { AcceptorConcentration DonorConcentration DopingConcentration
      TotalConcentration BoronActiveConcentration BoronConcentration
      BoronMinusConcentration PhosphorusActiveConcentration
      PhosphorusConcentration PhosphorusPlusConcentration eDensity hDensity
      eMobility hMobility BuiltinPotential ElectricField ElectrostaticPotential Space-
      Charge SRHRecombination Band2BandGeneration TotalRecombination eCur-
      rentDensity hCurrentDensity TotalCurrentDensity BandGap BandgapNarrowing
      eDriftVelocity }
Solve { * Получение начального решения:
      Poisson
      Coupled { Poisson Electron Hole }

      * Решение системы уравнений в квазистационарных
      *условиях при увеличивающемся напряжении на затворе
      QuasiStationary ( InitialStep = 0.001 MaxStep = 0.01 MinStep = 1e-7
                      Goal { name = "gate" voltage = 5 })
                      { Coupled {Hole Electron Poisson} } }
```

Рассчитанная передаточная вольт-амперная характеристика n -MOS-транзистора приведена на рисунке 6.1.

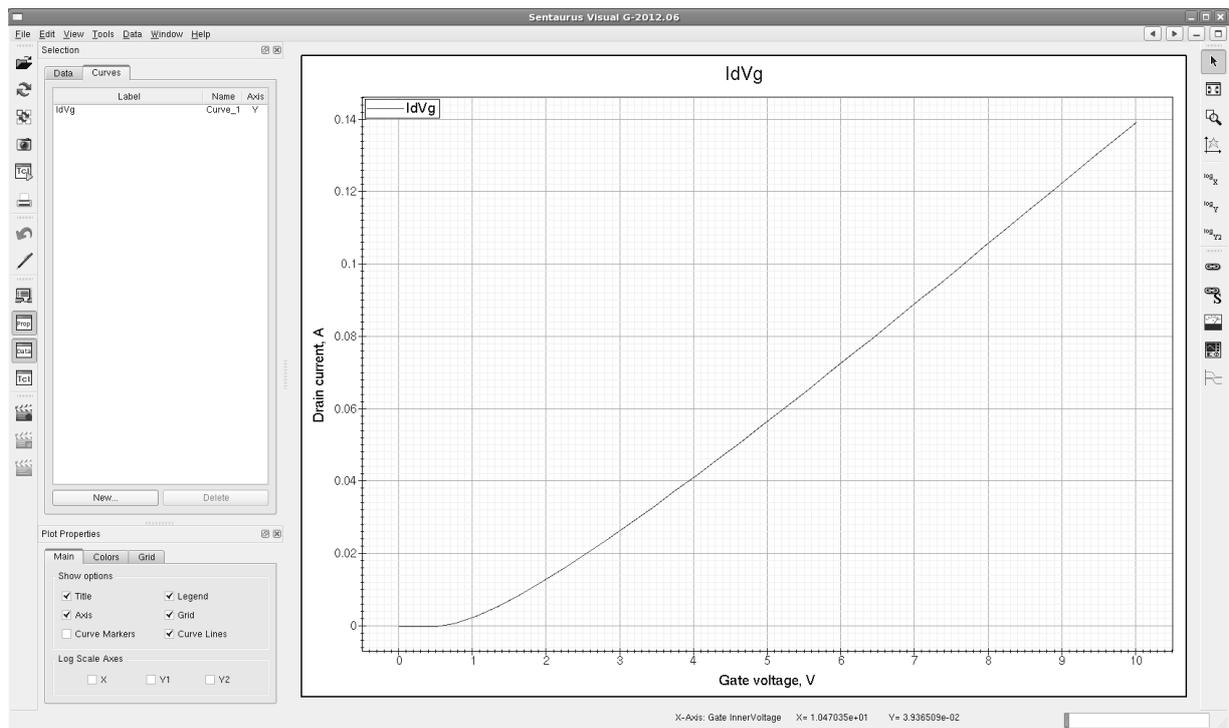


Рис. 6.1. Рассчитанная с помощью модуля *SDevice* передаточная вольт-амперная характеристика n -MOS-транзистора

7. ПРОГРАММА INSPECT ДЛЯ ПРОСМОТРА И ОБРАБОТКИ ГРАФИКОВ

Программа **Inspect** служит для построения и анализа графиков. **Inspect** имеет простой графический интерфейс. Имеется возможность управления процессом построения и анализа графиков с помощью командного файла. Графическое окно программного модуля **Inspect** показано на рисунке 7.1.

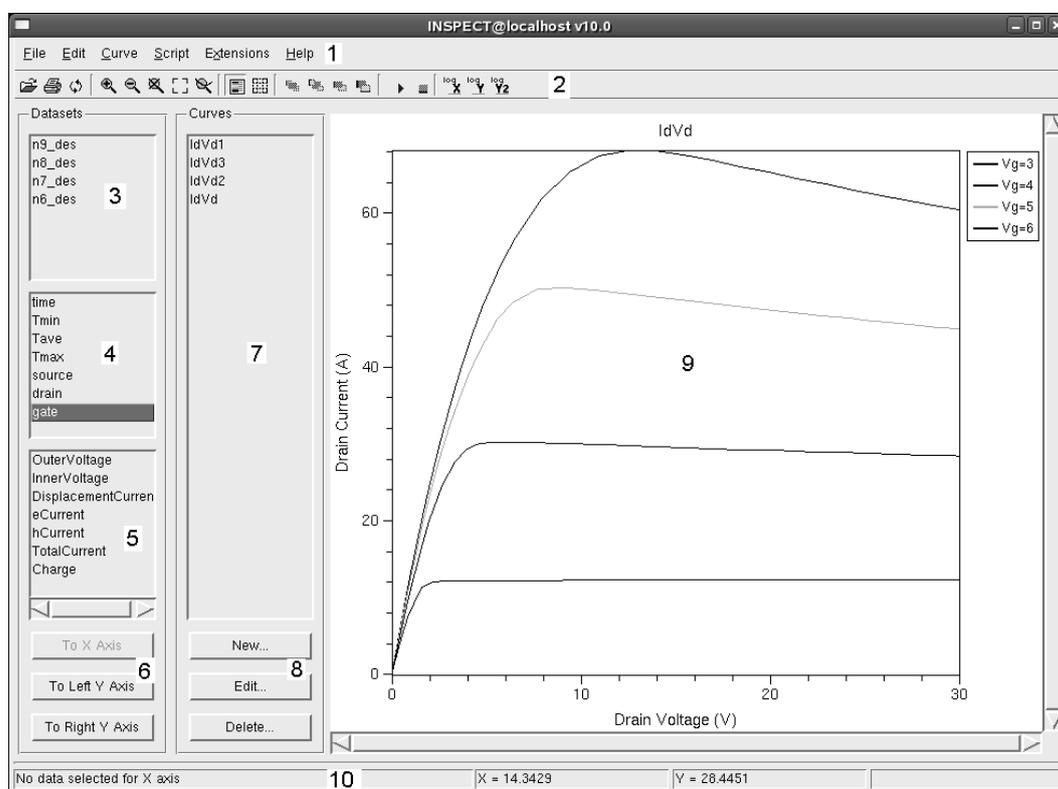


Рис. 7.1. Графическое окно программы **Inspect**:

1 – строка меню; 2 – строка кнопок управления; 3 – список загруженных файлов; 4 – область электродов, термодов, времени и т.п.; 5 – область параметров; 6 – кнопки выбора оси; 7 – список построенных графиков; 8 – кнопки создания и редактирования графиков; 9 – область графиков; 10 – строка состояния

Из пунктов строки меню наиболее используемые **File --> Load Datasets** и **Edit --> Plot Area (Axes)**. **Inspect** может загружать файлы с расширением *.plt, *.plx и др. (рис. 7.2).

Список загруженных файлов отображается в соответствующем поле области **Datasets** окна программы (рис. 7.1, поз. 3).

Назначение большинства кнопок управления (рис. 7.1, поз.2) интуитивно понятно и не требует пояснения. Кнопка **Reload** служит для обновле-

ния содержимого загруженных файлов при динамическом отображении графиков, например, в процессе расчета в **SDevice**. Можно также включить автоматическое обновление: **File** → **Automatically Update Datasets**. Для увеличения какой-либо области графика нужно, нажав кнопку **Zoom In**, выделить указателем мыши увеличиваемую область. Для уменьшения графика нужно нажать кнопку **Zoom Out**. Для приведения графика к исходному виду необходимо нажать кнопку **Zoom Off**. Используя кнопки **log X**, **log Y**, **log Y2** можно установить логарифмические координаты на соответствующих осях.

В поле электродов, термодов и др. (рис. 7.1, поз. 4) области **Datasets** отображаются контакты или другие элементы, существующие в загруженном файле. В поле параметров (рис. 7.1, поз. 5) отображаются параметры для выбранного контакта. Кнопки выбора оси (рис. 7.1, поз. 6) служат для привязки выбранного параметра к координатной оси.

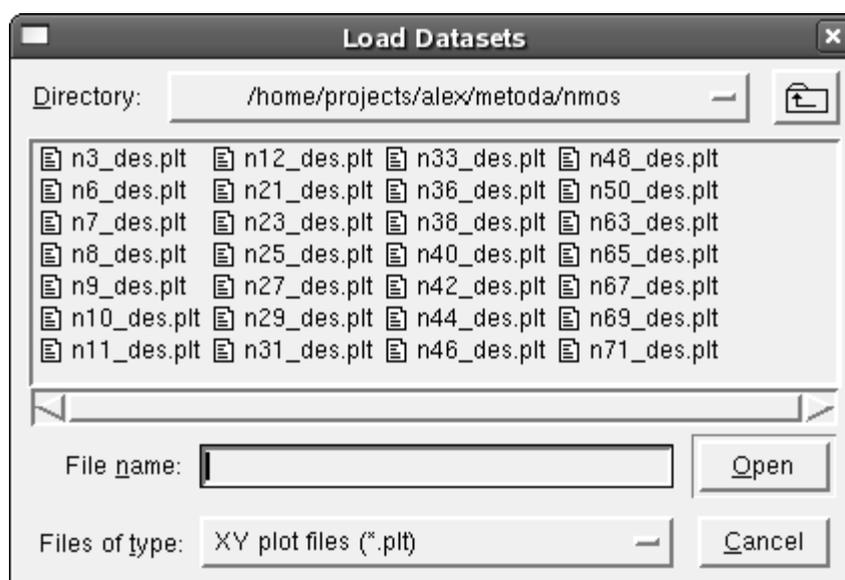


Рис. 7.2. Окно **Load Datasets** загрузки файлов с данными для построения графиков в программе **Inspect**

Для построения графика по данным из загруженного файла *.plt необходимо выполнить следующие действия: выделить нужный загруженный файл с данными; выделить нужный контакт; выделить параметр для данного контакта и выбрать ось, на которой будет отображаться диапазон значений данного параметра. При этом первым нужно задавать параметр по оси X. Такие же действия проделать для задания параметров по осям Y и Y2. Таким же образом строятся различные вольт-амперные характеристики и другие подобные зависимости.

Файл с расширением *.plx достаточно просто загрузить, и график будет построен автоматически. Следует учесть, что для просмотра профилей распределения примесей из **SProcess** нужно использовать полулогарифмические координаты: $X - \log Y$.

Например, для построения передаточной характеристики MOS-транзистора нужно напряжение на затворе *gate* -> **InnerVoltage** отложить по оси X , а полный ток стока *drain* -> **TotalCurrent** – по оси Y .

Список построенных кривых отображается в области **Curves** графического окна **Create Curve** (рис. 7.1, поз. 7). Кнопки внизу данной области позволяют вести математическую обработку кривых (кнопка **New**) и редактировать параметры отображения графиков. Появится окно обработки кривой (рис. 7.3).

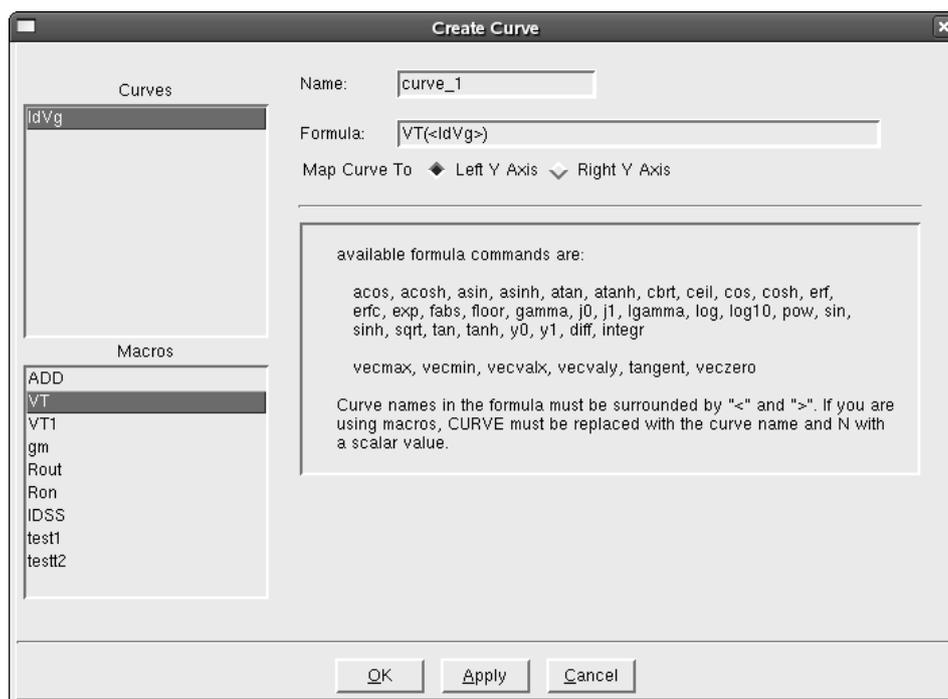


Рис. 7.3. Окно **Create Curve** обработки графика и расчета некоторых параметров

Для обработки графиков в **Inspect** имеется достаточно большой набор средств, начиная от простейших тригонометрических формул и функций вычисления производных до функций для фурье-анализа.

Помимо формул **Inspect** предоставляет набор predefined макросов. Наиболее часто используемые макросы:

ADD – сложение двух графиков;

VT – расчет порогового напряжения MOS-транзистора;

V_{T1} – расчет порогового напряжения по уровню тока стока 0.1 мкА;
 g_m – расчет крутизны передаточной характеристики MOS-транзистора.

Макросы используются следующим образом: в поле **Macros** выбрать нужный макрос, в поле **Curves** выбрать необходимый график или графики для выполнения макроса ADD сложения графиков.

При нажатии на кнопку **Edit** области **Curves** графического окна вызывается окно **Curve Attribute** настройки внешнего вида и атрибутов графика (рис. 7.4).

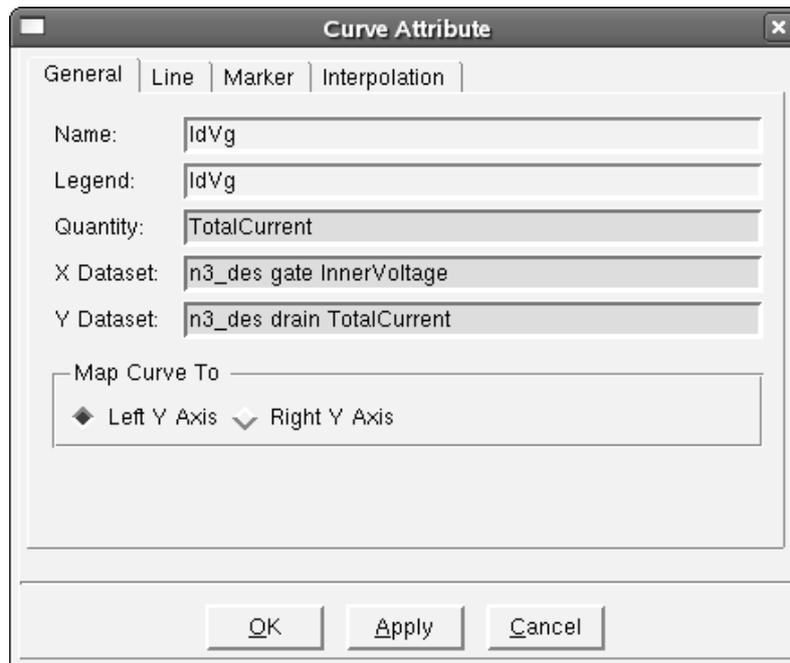


Рис. 7.4. Окно **Curve Attribute** настройки внешнего вида и атрибутов графика

Для редактирования внешнего вида области графиков и формата координатных осей используются пункты **меню Edit -> Plot Area** (рис. 7.5) и **Edit -> Axes** (рис. 7.6).

В качестве примера ниже приведен командный файл **Inspect**, используемый для построения передаточной вольт-амперной характеристики $I_d(V_g)$ n -MOS-транзистора, а также для расчета порогового напряжения V_T и крутизны S (листинг 7.1).

Листинг 7.1

```
#-----#  
gr_setTitleAttr "IdVg"  
#-----#
```

```

## Загрузка данных:
set dset @plot@
set data [file rootname $dset]
proj_load $dset

## Создание и отображение
## передаточной вольт-амперной характеристики  $I_d(V_g)$ :
cv_create IdVg "$data gate InnerVoltage" "$data drain TotalCurrent" y

## Отображение кривой:
cv_display IdVg y

## Установка атрибутов кривой:
cv_setCurveAttr IdVg "IdVg" black solid 2 none 3 defcolor 1 defcolor

## Установка атрибутов координатных осей:
gr_setAxisAttr X {Gate Voltage (V)} 12 {} {} black 1 12 0 5 0
gr_setAxisAttr Y {Drain Current (A)} 12 {} {} black 1 12 0 5 0
gr_setAxisAttr Y2 {S (A/V)} 12 {} {} black 1 12 0 5 0

## Создание и отображение зависимости крутизны  $S$ 
## от напряжения  $V_g$  на затворе:
cv_createWithFormula S_Vg "diff(<IdVg>)" A A A A
cv_setCurveAttr S_Vg "S(Vg)" green solid 2 none 3 defcolor 1 defcolor
cv_display S_Vg y2

## Расчет порогового напряжения  $V_T$  по уровню тока стока  $0.1 \text{ мкА}$ :
set VT [cv_compute "vecvalx(<IdVg>, 0.1)" A A A A]

## Запись переменной:
ft_scalar VT $VT

## Расчет максимальной крутизны характеристики:
set S [cv_compute "vecmax(<S_Vg>)" A A A A]
ft_scalar S $S

## Расчет максимального тока стока  $I_{max}$ :
set Imax [cv_compute "vecmax(<IdVg>)" A A A A]
ft_scalar Ids_max $Imax

```

В результате выполнения командного файла (листинг 7.1) были рассчитаны электропараметры транзисторной n -MOS-структуры: пороговое напряжение $V_T = 2,10 \text{ В}$; крутизна $S = 2,89 \text{ мСм}$; максимальный ток стока $I_{max} = 12,50 \text{ мА}$.

На рисунке 7.7 приведены построенные с помощью командного файла 7.1 передаточная вольт-амперная характеристика $I_d(V_g)$ и зависимость крутизны от напряжения на затворе $S(V_g)$.

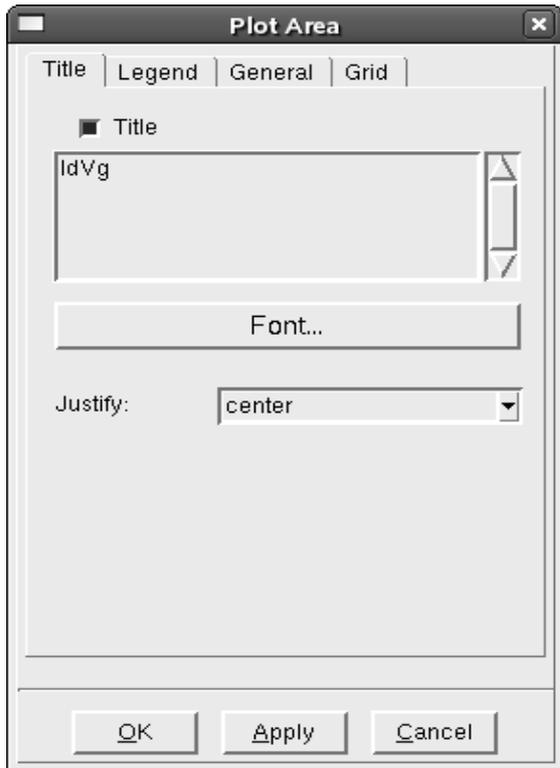


Рис. 7.5. Окно **Plot Area** настройки внешнего вида области графика

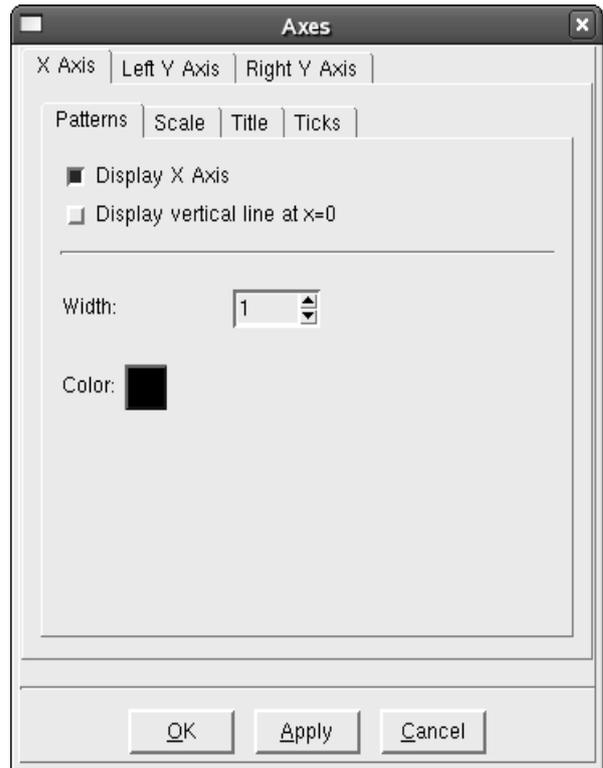


Рис. 7.6. Окно **Axes** настройки атрибутов координатных осей

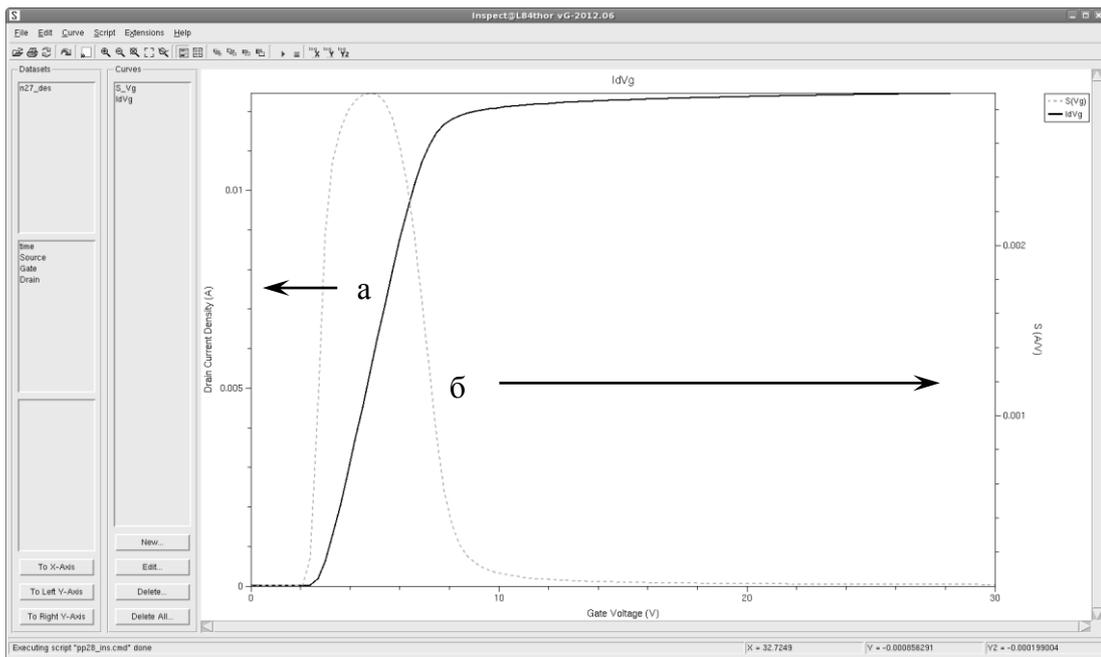


Рис. 7.7. Построенные с помощью командного файла 7.1 характеристики *n*-MOS-структуры:

- a* - — передаточная вольт-амперная характеристика $I_d(V_g)$;
- б* - зависимость крутизны от напряжения на затворе $S(V_g)$

8. УНИВЕРСАЛЬНАЯ ПРОГРАММА TECPLOT SV ДЛЯ ПРОСМОТРА РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ

Tecplot SV – программа для визуального представления результатов расчетов, например, распределений концентрации примесей, плотностей токов, подвижностей носителей и т.д. **Tecplot SV** может работать с одно-, двух- и трехмерными структурами и обладает очень большими возможностями визуализации. Окно программы **Tecplot SV** показано на рисунке 8.1.

При загрузке файлов структуры необходимо учитывать, что **Tecplot SV** использует пару файлов *.dat и *.grd.

Наиболее часто используемые пункты строки меню:

File -> Load – загрузка данных (рис. 8.2);

View -> Maximize Workspace – увеличение области просмотра на все окно;

Axis – настройка внешнего вида координатных осей (рис. 8.3);

Plot -> Contour – изменение количества изолиний, режима цветового градиента, меток изолиний и т.д. для выбранного параметра (рис. 8.4);

Slicer – построение сечений;

Data -> Probe At – отображение значений всех параметров в заданной точке (рис. 8.5).

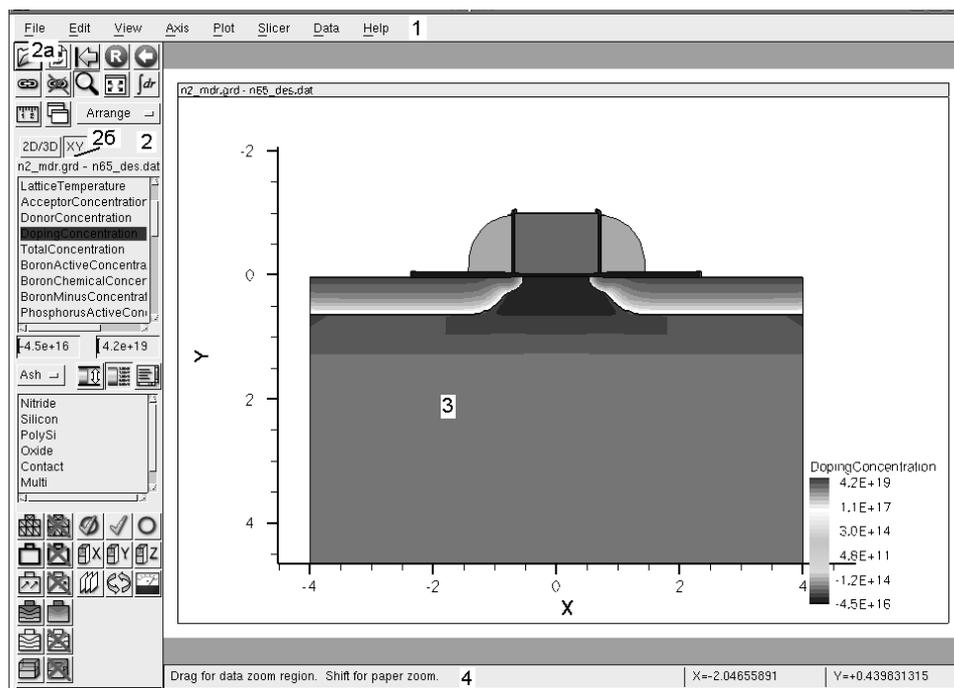


Рис. 8.1. Окно программы **Tecplot SV**:
1 – строка меню; 2 – панель управления;
3 – область просмотра; 4 – строка состояния

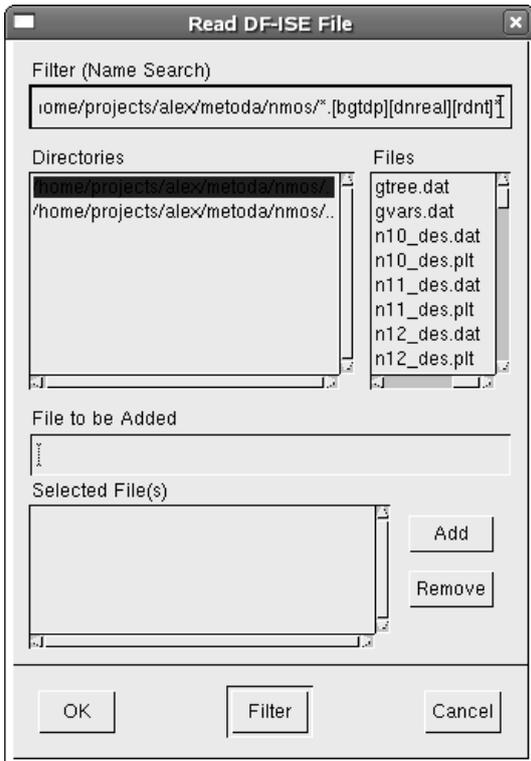


Рис. 8.2. Окно загрузки файлов в Tecplot SV

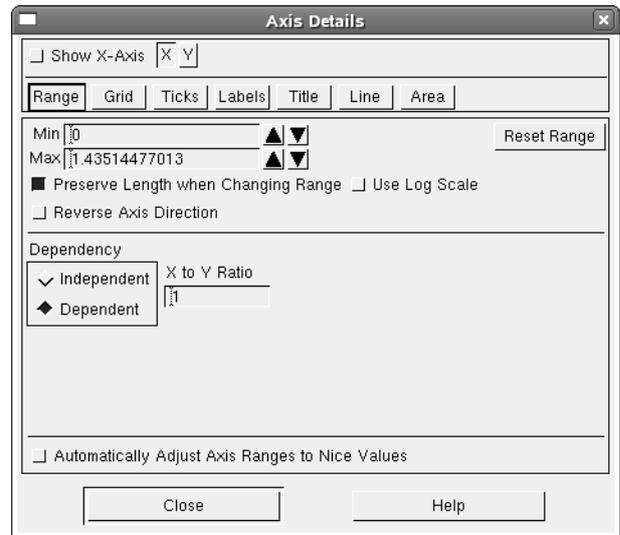


Рис. 8.3. Окно настройки атрибутов координатных осей Tecplot SV

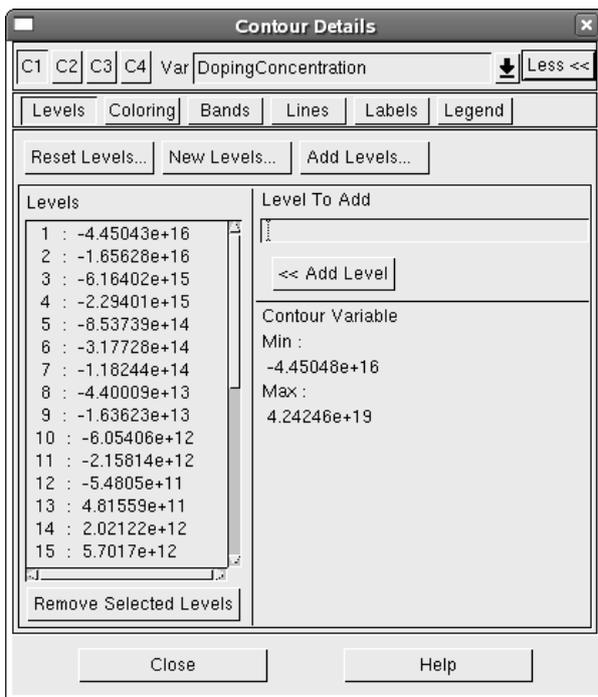


Рис. 8.4. Окно настройки изолиний, меток на них, цветового градиента и других установок Tecplot SV

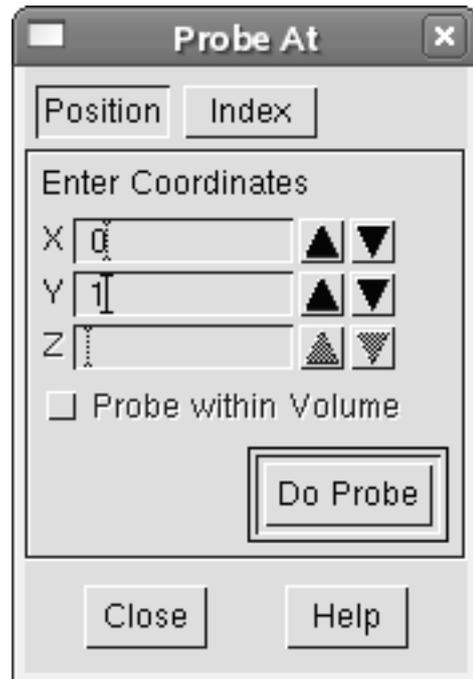


Рис. 8.5. Окно задания координат точки для получения значений всех её параметров в Tecplot SV

Панель управления можно разделить на две части: кнопки управления, используемые в любом режиме просмотра, и область, вид которой зависит от режима просмотра – одномерный или многомерный.

Основные кнопки управления (рис. 8.1, 2а):



– **Load** – загрузка данных;



– **Reset** – приведение к начальному виду;



– **Redraw** – перерисовка изображения;



– **Last View** – предыдущий вид;



– **Zoom** – увеличение изображения;



– **Measure** – измерение расстояния между двумя и более точками.

Ниже кнопок управления находится переключатель режима просмотра (рис. 8.1, 2б) – одномерный или многомерный (двух- или трехмерный). Одномерный режим используется для просмотра графиков. Далее рассматривается только двухмерный режим просмотра (рис. 8.6).

После загрузки данных в области просмотра отображается исследуемая структура. Для увеличения какой-либо ее области нужно нажать кнопку **Zoom** и выделить левой кнопкой мыши рассматриваемую прямоугольную область. Для перемещения структуры нужно, нажав и удерживая правую кнопку мыши, перемещать курсор в нужную сторону. Для измерения расстояния между двумя точками, а также измерения периметра и т.п. нужно нажать кнопку **Measure**, щелчком левой кнопки мыши задать начальную точку, далее, указывая щелчком левой кнопки мыши узловые точки, провести всю линию. Конечная точка обозначается двойным щелчком мыши (рис. 8.7).

Для включения/выключения сетки на каком-либо материале или области нужно сначала выбрать материал/область, затем нажать кнопку включения/выключения сетки. Аналогично производится включение/выключение отображения границ, изолиний и т.п.

Цифрами на рисунке 8.6 обозначены:

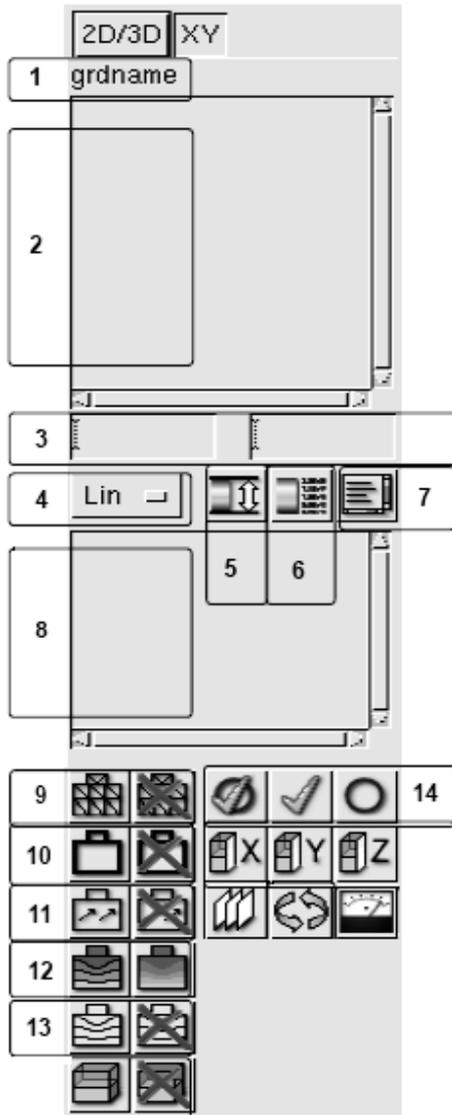


Рис. 8.6. Панель управления программы **Tecplot SV** в многомерном режиме

1 – имена используемых файлов структуры;

2 – список параметров просмотра, в нем отображаются те величины, распределения которых можно просмотреть для данной структуры (например, распределение примесей);

3 – поля минимального и максимального значений выбранного параметра;

4 – кнопка переключения режима интерполяции при отображении (линейная, логарифмическая, с использованием гиперболического арксинуса);

5 – кнопка выбора различных режимов вычисления минимального и максимального значения параметров;

6 – кнопка включения/выключения отображения цветовой шкалы значений параметра;

7 – кнопка переключения режимов отображения списка материалов/областей;

8 – список материалов либо областей;

9 – кнопка включения/выключения отображения сетки выбранного материала/области;

10 – кнопка включения/выключения отображения границ выбранного материала/области;

11 – то же для поля векторов;

12 – кнопка включения/выключения отображения изолиний при включенном цветовом градиенте;

13 – кнопка включения/выключения отображения изолиний при выключенном цветовом градиенте;

14 – кнопки режима отображения выбранного материала/области (слева направо): отображение только выбранного материала, отображение выбранного материала, выключение отображения выбранного материала

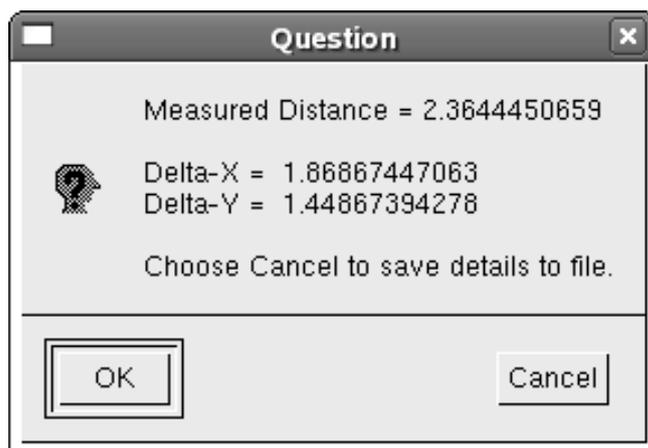


Рис. 8.7. Результат измерения расстояния в программе **Tecplot SV**

Управление **Tecplot SV** можно осуществлять с помощью командного файла. Ниже в качестве примера приведен командный файл **Tecplot SV** (листинг 8.1), предназначенный для отображения рассчитанной ранее структуры.

Листинг 8.1

```
#rem#!MC 800
$!FRAMECONTROL DELETETOP
$!DRAWGRAPHICS NO
$!READDATASET "@grid@ @doping@" DATASETREADER = "DF-ISE Loader"
$!VIEW FIT
$!CREATENEWFRAME XYPOS {X = 4.0 Y = 1.0} WIDTH = 3.0 HEIGHT = 4.0
$!READDATASET "@grid@ @doping@" DATASETREADER = "DF-ISE Loader"
$!GETVARNUMBYNAME |doping| NAME = "DopingConcentration"
$!GLOBALCONTOUR VAR = |doping|
$!FRAMELINKING LINKGROUP = 2
$!FRAMECONTROL POP FRAME = 1
$!VIEW FIT
$!VIEW ZOOM X1 = -4.0 Y1 = -2.3 X2 = 4.0 Y2 = 4.0
$!REDRAWALL
```

9. УНИВЕРСАЛЬНАЯ ПРОГРАММА ОТОБРАЖЕНИЯ ИНФОРМАЦИИ SENTAURUS VISUAL

Как и программа **Tecplot SV**, модуль **SENTAURUS Visual (SVisual)** способен и отображать смоделированную структуру, и строить графики различных зависимостей. Благодаря своему широкому функционалу **SVisual** способен заменить и **Tecplot SV**, и **Inspect**. В связи с этим **SVisual** может считаться базовой программой отображения информации в САПР **SENTAURUS**. Вид графического окна **SVisual** при отображении структуры представлен на рисунке 9.1.

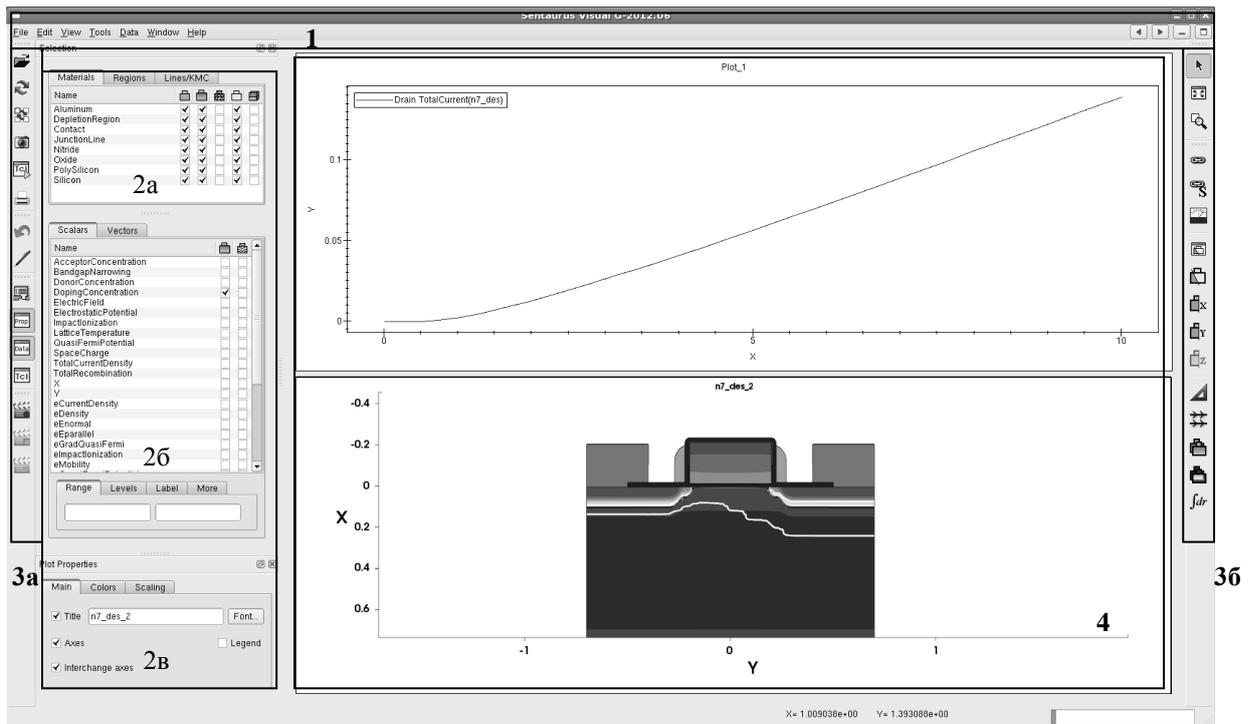


Рис. 9.1. Окно программы **SVisual** при работе со структурами:
1 – строка меню; 2 – панель управления;
3 – боковые панели управления; 4 – область просмотра

Строка меню (рис. 9.1, поз. 1) в целом идентична меню **Tecplot SV** и **Inspect**.

Панель управления (рис. 9.1, поз. 2) разделена на три отдельных «окна». Верхнее из них (рис. 9.1, поз. 2a) позволяет работать с элементами модели структуры. Элементы распределены по трем вкладкам: **Materials** – материалы; **Regions** – отдельные области; **Lines/KMC** – линии и контакты. Для каждого отдельного элемента можно включить/выключить: отображение элемента; отображение выбранного поля; отображения сетки; отобра-

жения контуров элемента; прозрачность. Второе окно (рис. 9.1, поз. 2б) включает поля распределения по структуре физических величин, например, концентрации примеси. В двух вкладках разграничены скалярные *Scalars* и векторные *Vectors* поля. Для каждого поля можно включить отображение через градиент цветов и/или соответствующий им набор изолиний. Цветовым градиентом может отображаться одновременно только одно поле, для изолиний такого ограничения нет. Нижнее окно (рис. 9.1, поз. 2в) панели управления позволяет настраивать саму область просмотра.

Боковые панели управления (рис. 9.1, поз. 3а и 3б) представлены набором кнопок, дублирующих функции пунктов строки меню. Наиболее часто используемые из них:

-  – увеличение изображения;
-  – «связывание» нескольких структур вместе, при этом работа со связанными структурами происходит синхронно;
-  – показ значения всех полей в выбранной точке;
-  – создание сечения вдоль оси *X* в выбранной координате;
-  – создание сечения вдоль оси *Y* в выбранной координате;
-  – измерение расстояния между двумя точками.

При работе с графиками вид окна **SVisual** изменяется (рис. 9.2).

Список загруженных файлов отображается во вкладке *Data* в верхней части окна программы (рис. 9.2, поз. 2). Вкладка *Curves* содержит уже построенные графики.

В области электродов, термодов и др. (рис. 9.2, поз. 3) находятся все контакты или другие элементы, существующие в загруженном файле. В области параметров (рис. 9.2, поз. 4) отображаются доступные параметры для выбранного контакта. В случае, если в качестве файла данных выбрано построенное сечение, область параметров будет пуста, а все доступные поля будут располагаться в области электродов. Кнопки выбора оси (рис. 9.2, поз. 5) служат для привязки выбранного параметра к координатной оси. Работа с ними проводится по аналогии с модулем **Inspect**. Кнопка *New Variable* аналогична кнопке *New* в **Inspect**, при ее нажатии всплывает окно *Create New Variable*, изображенное на рисунке 9.3.

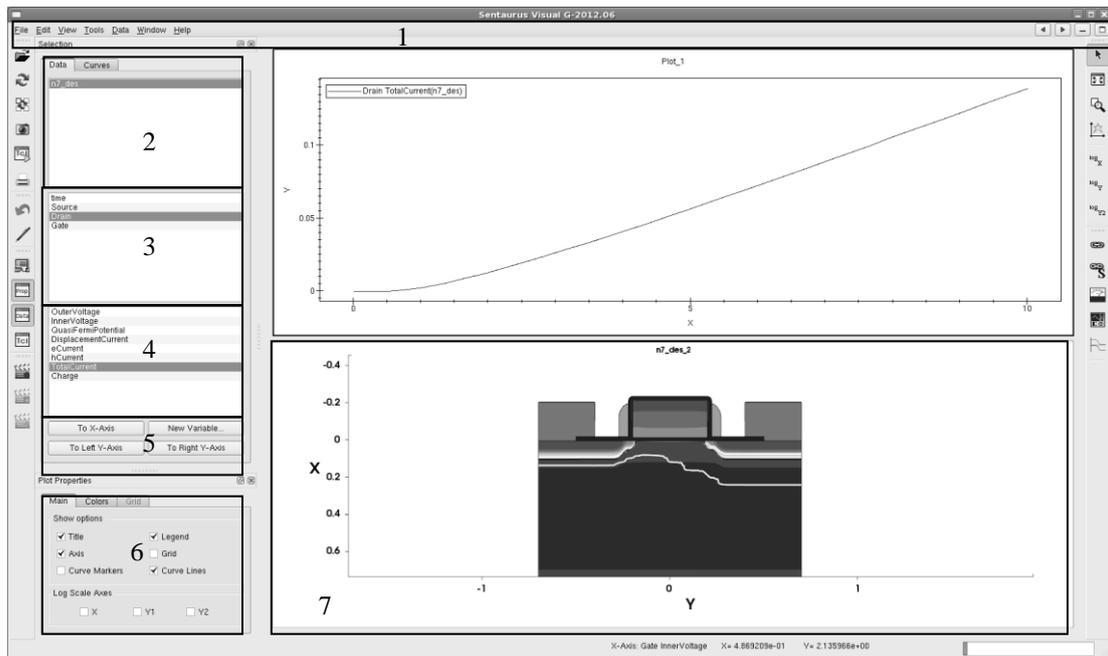


Рис. 9.2. Окно программы SVisual при работе с графиками: 1 – строка меню; 2 – список загруженных файлов; 3 – область электродов, термодов, времени и т.п.; 4 – область параметров; 5 – кнопки выбора оси; 6 – панель настройки области построения графика; 7 – область построения графика

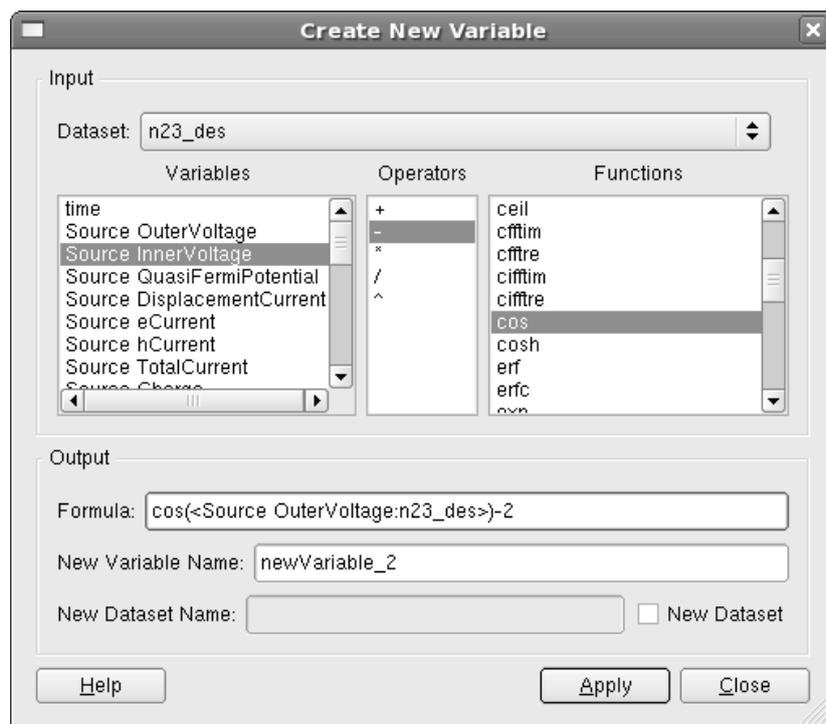


Рис. 9.3. Окно Create New Variable обработки графика в SVisual

Панель настройки области построения графика (рис. 9.2, поз. б) позволяет настраивать как всю область целиком, так и отдельные ее элементы, включая оси, заголовки и сами линии графика. Для области графика можно установить логарифмические координаты на той или иной оси, включить/выключить отображение осей, названия, сетки делений графика, маркеры линий, шрифт и размер заголовка и т.д. . Для линии графика можно поменять ее тип, толщину, цвет и т.д. Для осей координат настройки позволяют изменять шаг делений, область задания/определения графика, заголовок и т.д.

SVisual может быть встроен в дерево проекта и управляться командным файлом. Ниже приведен пример командного файла (листинг 9.1), аналогичный по назначению файлу, указанному в листинге 7.1 для **Inspect**.

Листинг 9.1

```

ext::SetInfoDef 1
#-----#
# Определение переменных
set N    @node@
set i    @node:index@
set Vd   @Vds@
set lo   @lth@
puts ""

# Присвоение цвета, привязанное к номеру узла
#-----#
set COLORS [list green blue red orange magenta violet brown]
set NCOLORS [length $COLORS]
set color  [index $COLORS [expr ${i%$NCOLORS}]]

#-----#
# Загрузка данных
load_file IdVg_@plot@ -name PLT($N)

# Создание графика передаточной вольт-амперной характеристики Id(Vg)
if {[lsearch [list_plots] Plot_IdVg] == -1} {
    create_plot -1d -name Plot_IdVg }
#Выбор созданного графика в качестве текущего
select_plots Plot_IdVg

# Определение массивов данных для расчета электропараметров
set Vgs [get_variable_data "Gate OuterVoltage" -dataset PLT($N)]
set Ids [get_variable_data "Drain TotalCurrent" -dataset PLT($N)]

```

```

# Создание кривой передаточной вольт-амперной характеристики  $I_d(V_g)$ 
create_curve -name IdVg($N) -dataset PLT($N) \
    -axisX "Gate OuterVoltage" -axisY "Drain TotalCurrent"

# Установка атрибутов кривой  $I_d(V_g)$ 
set_curve_prop IdVg($N) -label "IdVg (Vd=$Vd)" \
    -color $color -line_style solid -line_width 3

# Установка атрибутов области построения графика
# передаточной вольт-амперной характеристики  $I_d(V_g)$ 
set_plot_prop -title "I<sub>d</sub>-V<sub>g</sub> Curve" -title_font_size 20
set_axis_prop -axis x -title {Gate Voltage [V]} \
    -title_font_size 16 -scale_font_size 14 -type linear
set_axis_prop -axis y -title {Drain Current [A]} \
    -title_font_size 16 -scale_font_size 14 -type linear
set_legend_prop -font_size 12 -location bottom_right -font_att bold

# Запись порогового напряжения, тока стока насыщения  $I_{max}$ 
# и крутизны  $S$  в Workbench
ext::ExtractVti          out= Vti    name= "Vti"    v= $Vgs i= $Ids io= $Io
ext::ExtractExtremum    out= Idmax   name= "Idsat"  x= $Vgs y= $Ids extremum= "max"
ext::ExtractGm          out= gm      name= "gm"    v= $Vgs i= $Ids

```

В результате выполнения командного файла (листинг 9.1) были рассчитаны электропараметры транзисторной n -MOS-структуры: пороговое напряжение $V_T = 2,10$ В; крутизна $S = 2,89$ мСм; максимальный ток стока $I_{max} = 12,46$ мА.

На рисунке 9.4 приведена построенная с помощью командного файла 9.1 передаточная вольт-амперная характеристика $I_d(V_g)$

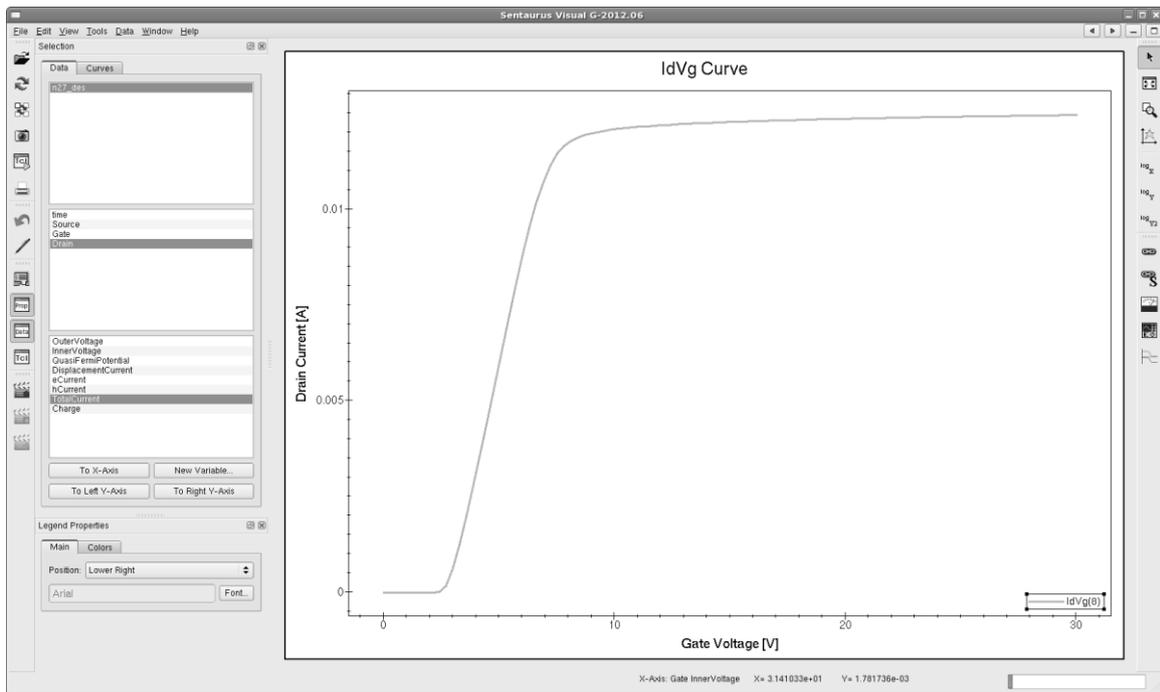


Рис. 9.3. Передаточная вольт-амперная характеристика $I_d(V_g)$ n -MOS-структуры, построенная в SVisual

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Глушко А. А. Приборно-технологическое моделирование в системе TCAD Sentaurus : учебно-методическое пособие / А. А. Глушко. – Москва : Издательство Московского государственного технического университета им. Н. Э. Баумана, 2015 . – 64 с.
2. Гуртов В. А. Твердотельная электроника / В. А. Гуртов. – Москва : Техносфера, 2007. – 408 с.
3. Зи С. Физика полупроводниковых приборов : в 2 кн. / С. Зи. – Москва : Мир, 1984. – Кн. 1. – 456 с.; Кн. 2. – 456 с.
4. Зыков Д. Д. Системы автоматизированного проектирования технологических процессов и технологических маршрутов производства СВЧ МИС, оптимизация производства (основы САПР Synopsys TCAD) : учебное пособие / Д. Д. Зыков, К. Ю. Осипов. – Томск : В-Спектр, 2010. – 76 с.
5. Королев М. А. Приборно-технологическое моделирование при разработке изделий микроэлектроники и микросистемной техники / М. А. Королев, Т. Ю. Крупкина, Ю. А. Чаплыгин. // Изв. вузов. Электроника. – 2005. – № 4-5. – С. 64-71.
6. Моделирование нанотранзисторов в TCAD Sentaurus : методическое руководство к лабораторному практикуму / С. В. Калинин [и др.]. – Новосибирск : Издательство Новосибирского государственного технического университета, 2010. – 100 с.
7. Нелаев В. В. Основы САПР в микроэлектронике: учебное пособие / В. В. Нелаев, В. Р. Сتمпицкий. – Минск : Белорусский государственный университет информатики и радиоэлектроники, 2008. – 221 с.
8. Пикулев В. Б. Основы моделирования планарных МДП-структур в физике твёрдого тела : учебное пособие / В. Б. Пикулев. – Петрозаводск : Издательство Петрозаводского государственного университета, 2005. – 74 с.
9. Приборно-технологическое проектирование полевых полупроводниковых приборов : учебно-методическое пособие / сост. : А. В. Быстрицкий [и др.]. – Воронеж : Издательский дом ВГУ, 2017. – 36 с.
10. Приборно-технологическое проектирование элементной базы мощной СВЧ-электроники : учебно-методическое пособие / сост. : Р.П. Алексеев [и др.]. – Воронеж : Издательский дом ВГУ, 2016. – 69 с.
11. Сайт фирмы Siborg System Inc. [Электронный ресурс]. – (<http://www.siborg.com>). – Дата обращения 12.03.2017.

12. Сайт фирмы Silvaco [Электронный ресурс]. – (<http://www.silvaco.com>). – Дата обращения 12.03.2017.
13. Сайт фирмы Synopsys [Электронный ресурс]. – (<http://www.synopsys.com/>) – Дата обращения 12.03.2017.
14. Технология, конструкции и методы моделирования кремниевых интегральных микросхем. Элементы и маршруты изготовления кремниевых ИС и методы их математического моделирования / М. А. Королёв [и др.]. – Москва : Бином. Лаборатория знаний, 2009. – Ч. 2. – 422 с.
15. Тихомиров П. Система Senraurus TCAD компании Synopsys / П. Тихомиров, П. Пфеффли, М. Зорзи. // Электроника: Наука. Технология. Бизнес. – 2006. – № 7. – С. 89-95.
16. Synopsys World Leader in EDA Software and Services. – (<http://www.synopsys.com>). – Дата обращения 12.03.2017.

Учебное издание

ОСНОВЫ РАБОТЫ
В СРЕДЕ ПРИБОРНО-ТЕХНОЛОГИЧЕСКОЙ САПР
SENTAURUS

Учебно-методическое пособие

Составители:

Алексеев Роман Павлович,
Бормонтов Евгений Николаевич,
Быкадорова Галина Владимировна,
Ткачёв Александр Юрьевич,
Цоцорин Андрей Николаевич

Издано в авторской редакции

Подписано в печать 09.08.2017. Формат 60×84/16
Уч.-изд. л. 5,2. Усл. печ. л. 5,6. Тираж 25 экз. Заказ 455

Издательский дом ВГУ
394018 Воронеж, пл. им. Ленина, 10

Отпечатано с готового оригинал-макета
в типографии Издательского дома ВГУ
394018 Воронеж, ул. Пушкинская, 3