

Министерство науки и высшего образования Российской Федерации

Национальный исследовательский университет «МЭИ»

Институт радиотехники и электроники им. В.А. Котельникова

Кафедра электроники и нанoeлектроники

Практикум по дисциплине

«Проектирование топологии компонентов интегральных схем»

для студентов, обучающихся на направлении 11.03.04 «Электроника и нанoeлектроника» по профилю «Микроэлектроника и твердотельная электроника»

Москва, 2022

Начало работы.....	3
Подготовка к подключению технологической библиотеки gpdk045.....	3
Запуск Virtuoso.....	4
Практическое занятие № 1. Создание библиотеки элементов и моделирование	6
Создание библиотеки.....	6
Создание схемы инвертора (schematic-представление).....	6
Создание символа инвертора (symbol-представление).....	9
Самостоятельная работа.....	11
Практическое занятие № 2. Создание топологического изображения логического элемента.....	12
Создание топологии логического элемента (layout-представление).....	12
Экстракция паразитных параметров схемы (av_extracted-представление). 20	
Создание конфигурационного файла.....	22
Самостоятельная работа.....	29
Практическое занятие № 3. Создание топологического изображения компонента.....	31
Самостоятельная работа.....	31
Практическое занятие № 4. Генерация LEF-файла.....	33
Самостоятельная работа.....	35
Практическое занятие № 5. Генерация LIB-файла.....	41
Практическое занятие № 6. Поток проектирования.....	44
Моделирование HDL-описания.....	44
Синтез логической схемы.....	45
Синтез с использованием скрипта.....	49
Размещение и трассировка.....	50
Верификация топологии.....	66
Экспорт набора фотошаблонов в формате GDSII.....	70
Аналоговое моделирование кристалла.....	70
ПРИЛОЖЕНИЕ А.....	72
ПРИЛОЖЕНИЕ Б.....	73
Часто возникающие вопросы.....	75

Начало работы

Cadence может запускаться лишь на Unix-платформах. В связи с этим не лишним будет перед началом работы ознакомиться с некоторыми часто используемыми командами Linux. Для большей подробности информацию о командах можно найти в Интернет.

Таблица 1 — Список основных команд bash

Команда	Действие
ls	Выводит список файлов в текущей директории (от англ. list)
pwd	Показать текущую директорию (от англ. print working directory)
cd	Изменить текущую директорию (от англ. change directory)
mv	Переместить файл в другую директорию (от англ. move)
cp	Копировать файл (от англ. copy)
rm	Удалить файл (от англ. remove)
mkdir	Создать директорию (от англ. make directory)

В настоящей лабораторной работе используется технологическая библиотека GPDК045 (generic process design kit, универсальный набор для проектирования технологических процессов) для 45-нм технологии, разработанный компанией Cadence Design Systems.

Подготовка к подключению технологической библиотеки gpdk045

- 1 Откройте терминал Linux
- 2 В папке Documents создайте директорию со своей фамилией

```
$ mkdir ~/Documents/[имя_директории]
```
- 3 Перейдите в эту директорию

```
$ cd ~/Documents/[имя_директории]
```
- 4 В ней создайте директорию с проектом Cadence и перейдите в неё

```
$ mkdir [название_проекта] && cd [название_проекта]
```
- 5 В директории с проектом создайте дополнительно папки cds, lvs, drc, simulation, doc, netlist

```
$ mkdir cds lvs drc simulation doc netlist
```
- 6 Перейдите в папку cds, в которой будет храниться Ваша библиотека

```
$ cd cds
```

Вся дальнейшая работа пользователя должна быть организована **в этой директории**.

- 7 В этой директории любым текстовым редактором создайте документ `cds.lib`, например

```
$ gedit cds.lib
```

- 8 В открывшемся документе добавьте определения на необходимые библиотеки, сохраните и закройте документ (пример команды)

```
INCLUDE /local/1/gpdk045_v_6_0
```

```
INCLUDE /local/2/gpdk045_v_6_0
```

Приготовления завершены, можно переходить к запуску Virtuoso.

Запуск Virtuoso

- 1 При необходимости просмотра списка установленного ПО САПР наберите команду

```
$ module avail
```

- 2 Загрузите модуль ПО САПР Cadence **IC**, а также модули симулятора **SPECTRE**, проведения физической верификации **PVS** (Phisycal Verification System) и экстракции паразитных параметров **EXT** (Extraction) (номера версий могут отличаться)

```
$ module load cadence/IC/06.18.040
```

```
$ module load cadence/SPECTRE/19.10.237
```

```
$ module load cadence/PVS/16.15.000
```

```
$ module load cadence/EXT/19.12.000
```

- 3 В директории с **Вашей библиотекой (в папке `cds`)** запустите Virtuoso в фоновом режиме, чтобы можно было продолжать работать в той же сессии терминала

```
$ virtuoso &
```

- 4 Для автоматизации выполнения пп. 1—3 можно в папке `cds` сформировать скрипт, подгружающий модули и запускающий `virtuoso`. Для этого

создайте файл с именем load_virtuso.sh и скпируйте в него следующее содержание

```
# #!/bin/bash
echo "Loading module Cadence for design IC..."
module load cadence/IC/06.18.040
echo "done"
echo "Loading module Cadence for simulation SPECTRE..."
module load cadence/SPECTRE/19.10.237
echo "done"
echo "Loading module Cadence for physical verification
PVS..."
module load cadence/PVS/16.15.000
echo "done"
echo "Loading module Cadence for extraction parasitic EXT..."
module load cadence/EXT/19.12.000
echo "done"

echo "Running Virtuoso in the background..."
virtuoso &
```

Сохраните файл и в следующий раз запускайте его из папки cds командой

```
$ sh load_virtuoso.sh
```

Практическое занятие № 1. Создание библиотеки элементов и моделирование

Создание библиотеки

- 1 Откройте **Tools > Library Manager**.
- 2 Создайте новую библиотеку **File > New > Library**.
- 3 Задайте имя библиотеки, нажмите ОК.
- 4 При выборе технологического файла укажите *Attach to technology library*, что позволит прикрепить GPDК045 к Вашей библиотеке. Обратите внимание, что это очень важный шаг, без него дальше ничего работать не будет.
- 5 В открывшемся меню выберите gpdk045.
- 6 Если в меню нет gpdk045, обратитесь к преподавателю.

Создание схемы инвертора (schematic-представление)

- 1 Создайте изображение ячейки, для этого в Library Manager в списке библиотек выберите свою библиотеку, затем **File > New > Cell view**.
- 2 В открывшемся меню укажите в поле Cell наименование инвертора INV, убедитесь, что поле View имеет значение *schematic*, а также в Type имеет значение *schematic*.
- 3 Добавьте элемент — *p*-канальный транзистор **Create > Instance** (гор. клавиша I), в открывшемся окне выберите библиотеку, в которой расположен нужный элемент gpdk045 (pmos1v). Разместите объект на схеме.
- 4 Добавьте аналогично *n*-канальный транзистор (nmos1v), а также элементы питания vdd и gnd (последние — из библиотеки analogLib).
- 5 Соедините проводами элементы **Create > Wire** (гор. клавиша W). Выведите провода для подсоединения пинов входа и выхода.
- 6 Добавьте входной и выходной пин **Create > Pin** (гор. клавиша P). В открывшемся окне можно выбрать различные типы пинов: входной input, выходной output, двунаправленный inputOutput или переключатель switch. Для схемы инвертора требуются только входной и выходной пины.
- 7 Проверьте и сохраните кнопкой Check and Save, чтобы удостовериться в отсутствие ошибок.

Моделирование (временной анализ)

Моделирование происходит в симуляторе под названием Spectre. Данное моделирование работает как для схмотехнического представления ячейки, так и для топологического. На текущем моменте мы работаем со схмотехническим изображением. Моделирование топологического представления происходит аналогично.

- 1 Запустите ADE (Analog Design Environment) L: **Launch** > **ADE L**.
- 2 При запросе загрузки лицензии нажмите Always.
- 3 В новом окне откройте **Setup** > **Simulator/Directory/Host** и удостоверьтесь, что выбран симулятор Spectre. также здесь можно указать папку, в которую будут сохраняться результаты моделирования.
- 4 Укажите библиотеку со SPICE-моделями **Setup** > **Model Libraries** (она должна быть выбрана автоматически).
- 5 Задайте параметры тестовых воздействий **Setup** > **Stimuli**, определите входные сигналы (Inputs) и параметры питания (Global Sources).

Задайте входной сигнал в виде симметричного импульса (function = pulse) от 0 до 1,1 В с периодом 2 нс, длительностью активного уровня сигнала 1 нс (для симметричного импульса можно не задавать этот параметр) и длительностью фронтов 0,1 нс. Глобальный сигнал питания vdd относительно глобального сигнала земли gnd! задайте постоянным (function = dc) величиной 1,1 В. Убедитесь, что сигналы активны (стоит галочка Enabled).

6 Выберите тип анализа transient **Analyses** > **Choose**. В открывшемся окне выберете tran. Укажите длительность анализа в строке Stop Time равную 10n. Точность укажите moderate. Не выбирайте Transient Noise. Отметьте Enabled.

7 Далее необходимо выбрать сигналы, которые будут отображаться на графике. Для этого откройте окно **Outputs** > **Setup** и выберите **From Design**. Выберите сигналы In и Out (порт) на схеме. Либо можно выбрать соответствующий проводник. Выбор пина элемента (например, стока транзистора) приведёт к тому, что сигналом будет ток, вытекающий из него.

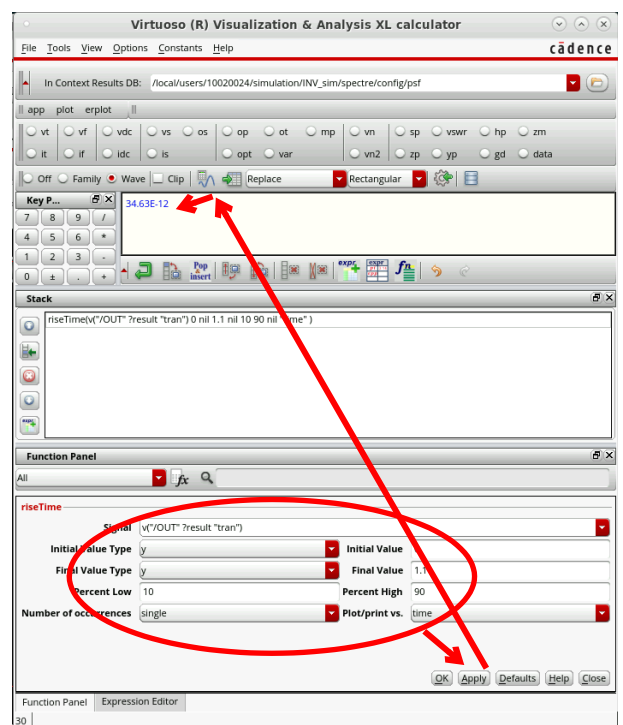
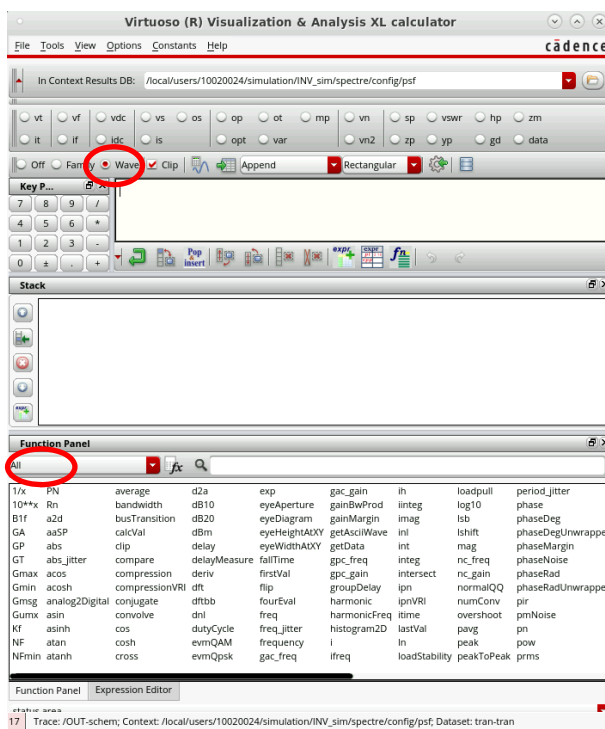
8 Убедитесь, что у выбранных сигналов стоит галочка Plot. Запустите моделирование **Simulation** > **Run**.

9 Появилось окно с результатами? Отлично!

Измерения

При анализе результатов моделирования часто необходимо произвести какие-либо измерения. Здесь это осуществляется следующим образом. Рассмотрим на примере расчёта времени нарастания сигнала по результатам моделирования реакции на импульс.

- 1 В окне меню **Tools > Calculator** выберите радиокнопку Wave.
- 2 В списке функций выберите All. Выберите опцию rise time.
- 3 Выберите сигнал из окна с осциллограммами, для которого необходимо рассчитать время фронта и нажмите ОК (кнопка может быть сразу не видна).
- 4 В окне настройки параметров расчёта убедитесь, что выбран выходной сигнал, а также расчёт производится по оси *y* в диапазоне от 0 до 1,1 В. Расчёт осуществляется по уровням 10 и 90 %.



- 5 Нажмите на буфер для вычислений, чтобы отобразить результат расчёта.

Моделирование (анализ по постоянному току)

Для проведения анализа по постоянному току (например, для получения передаточной характеристики логического элемента) на схему необходимо добавить источник постоянного тока.

- 1 Добавьте компонент источника постоянного тока *vdc* из библиотеки *analogLib*, у которого укажите параметр *DC voltage = 1.5 V*.

- 2 Удалите пин In, подключите положительный узел vdc ко входу инвертора, а к отрицательному подключите землю (gnd из библиотеки analogLib).
- 3 Проверьте схему кнопкой Check and Save.
- 4 Запустите ADE (Analog Design Environment) L: **Launch** > **ADE L**.
- 5 При запросе подгрузки лицензии нажмите Always.
- 6 В новом окне откройте **Setup** > **Simulator/Directory/Host** и удостоверьтесь, что выбран симулятор Spectre. Также здесь можно указать папку, в которую будут сохраняться результаты моделирования.
- 7 Укажите библиотеку со SPICE-моделями **Setup** > **Model Libraries** (она должна быть выбрана автоматически).
- 8 Задайте параметры тестовых воздействий **Setup** > **Stimuli**, определите параметры питания (Global Sources).

Глобальный сигнал питания задайте постоянным (function = dc) величиной 1,1 В. Убедитесь, что сигнал активен (стоит галочка Enabled).

- 9 Выберите тип анализа **Analyses** > **Choose**. В открывшемся окне выберете dc. В блоке Sweep variable укажите, по чему будет производиться развёртка графика — параметр компонента Component parameter. Кнопкой Select component выберите на схеме источник постоянного напряжения и укажите величину, которая будет изменяться — dc.

Диапазон изменения напряжения в блоке Sweep range выставьте от Start = 0 до Stop = 1.5. Количество точек укажите 1000 (или шаг 0,001 В).

- 10 Далее необходимо выбрать сигналы, которые будут отображаться на графике. Для этого выберите **Outputs** > **To Be Plotted** > **Select On Design**. Выберите порт Out на схеме.
- 11 Запустите моделирование **Simulation** > **Run**.
- 12 Появилось окно с результатом? Отлично!

Создание символа инвертора (symbol-представление)

Как известно, при проектировании цифровых схем широко используется принцип иерархичности. Перед созданием символа следует немного изменить схемотехническое представление компонента. Для этого сперва необходимо удалить элементы vdd и gnd, а на их место добавить входные пины VDD и GND. Далее можно создавать символ.

- 1 Создать новую ячейку, для этого в окне с графическим представлением схемы выберите **Create > CellView > From Cellview**.
- 2 Убедитесь, что имя ячейки INV.
- 3 В окне настройки вида символа разместите входной порт IN слева, а выходной OUT справа. Пины питания и земли VDD и GND расположены сверху и снизу соответственно. Нажмите ОК.
- 4 В новом окне можно изменить внешний вид символа. Обратите внимание, что красная граница - это невидимая область элемента, на которой будут располагаться пины, она имеет форму прямоугольника. Она определяет границу символа при его выделении на схемах. Зелёная граница - это изображение компонента, то есть то, что будет отображаться на схеме.
- 5 Теперь в Library Manager у ячейки INV можно увидеть несколько её представлений: схематехнический schematic (на транзисторах и иерархических блоках), символьный symbol (условно-графическое изображение), позже добавятся ещё топологический layout (фотошаблоны) и экстрагированный (SPICE-описание, экстрагированное из топологии).

Моделирование символьного представления (временной анализ)

- 1 В библиотеке компонентов создайте ячейку с названием INV_sim.
- 2 В схематехническом представлении добавьте символ инвертора.
- 3 Из библиотеки analogLib добавьте элементы gnd и vdd и соедините их с пинами GND и VDD соответственно.
- 4 Так же добавьте два источника питания - vdc и vpulse.
- 5 У источника vdc отрицательный узел подключите к gnd, а положительный — к элементу vdd. Значение напряжения у vdc выставьте как VDD (будет использоваться в качестве параметра при моделировании).
- 6 Параметры импульсного источника выставьте как для первого моделирования, только вместо высокого уровня сигнала укажите VDD.
- 7 На неподсоединённый выходной провод поставьте из библиотеки basic элемент noConn (указываем, что нет соединения).
- 8 Для моделирования запустите ADE L, в котором в списке параметров укажите, что VDD = 1,1 В.
- 9 Проведите моделирование по времени (реакция на импульс).

Самостоятельная работа

- 1 Создайте дополнительно две ячейки: 2И-НЕ (NAND2) и 2ИЛИ-НЕ (NOR2). Проведите для них моделирование и продемонстрируйте верные осциллограммы.
- 2 Создайте остальные базовые вентили, которые используются в Вашем проекте (schematic и symbol).

Содержание отчёта

Отчёт должен содержать:

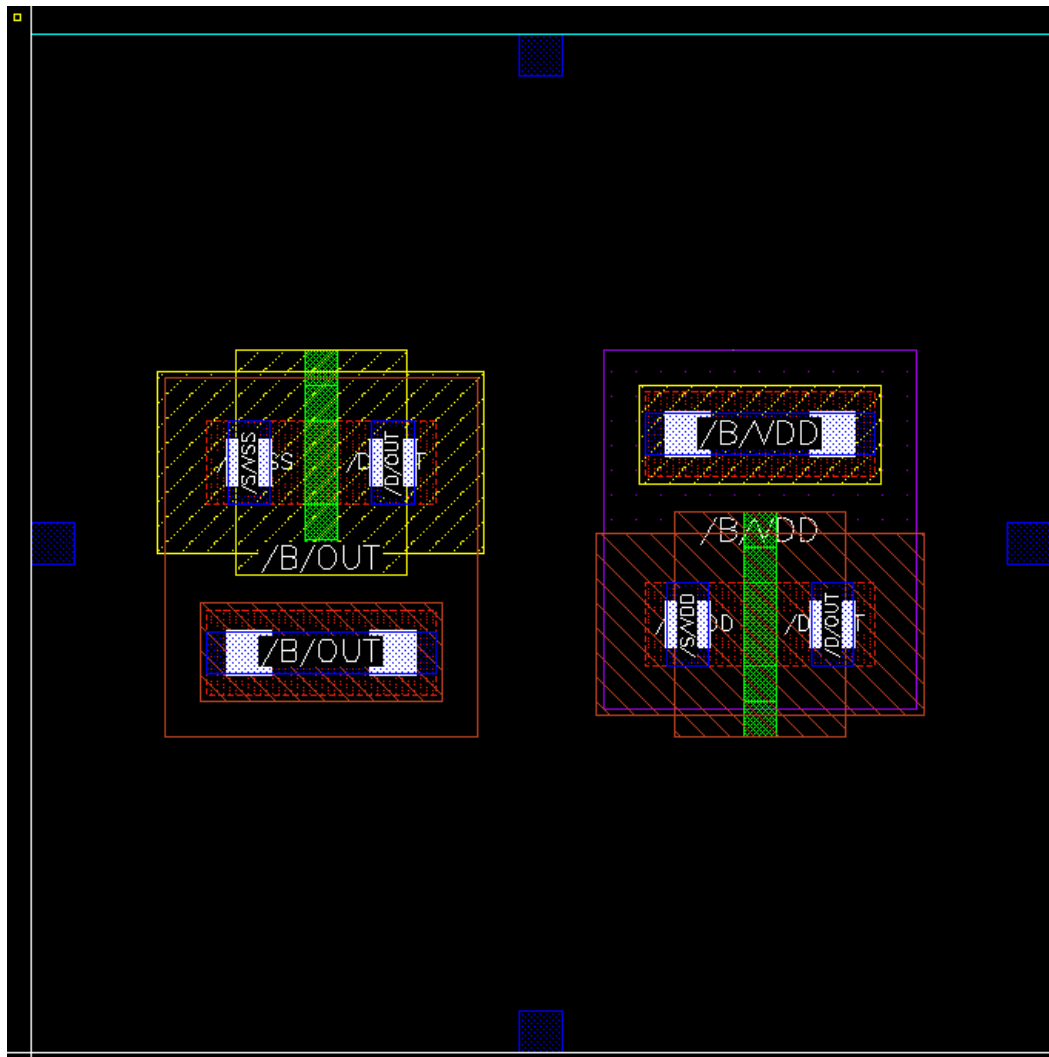
- 1 Титульный лист.
- 2 Задание.
- 3 Комментарии к выполнению.
- 4 Схемотехническое и символьное изображения логических элементов.
- 5 Результаты моделирования: реакцию на импульсное воздействие и передаточную характеристику инвертора, реакцию на импульсное воздействие ячеек 2И-НЕ и 2ИЛИ-НЕ.

Практическое занятие № 2. Создание топологического изображения логического элемента

Создание топологии логического элемента (layout-представление)

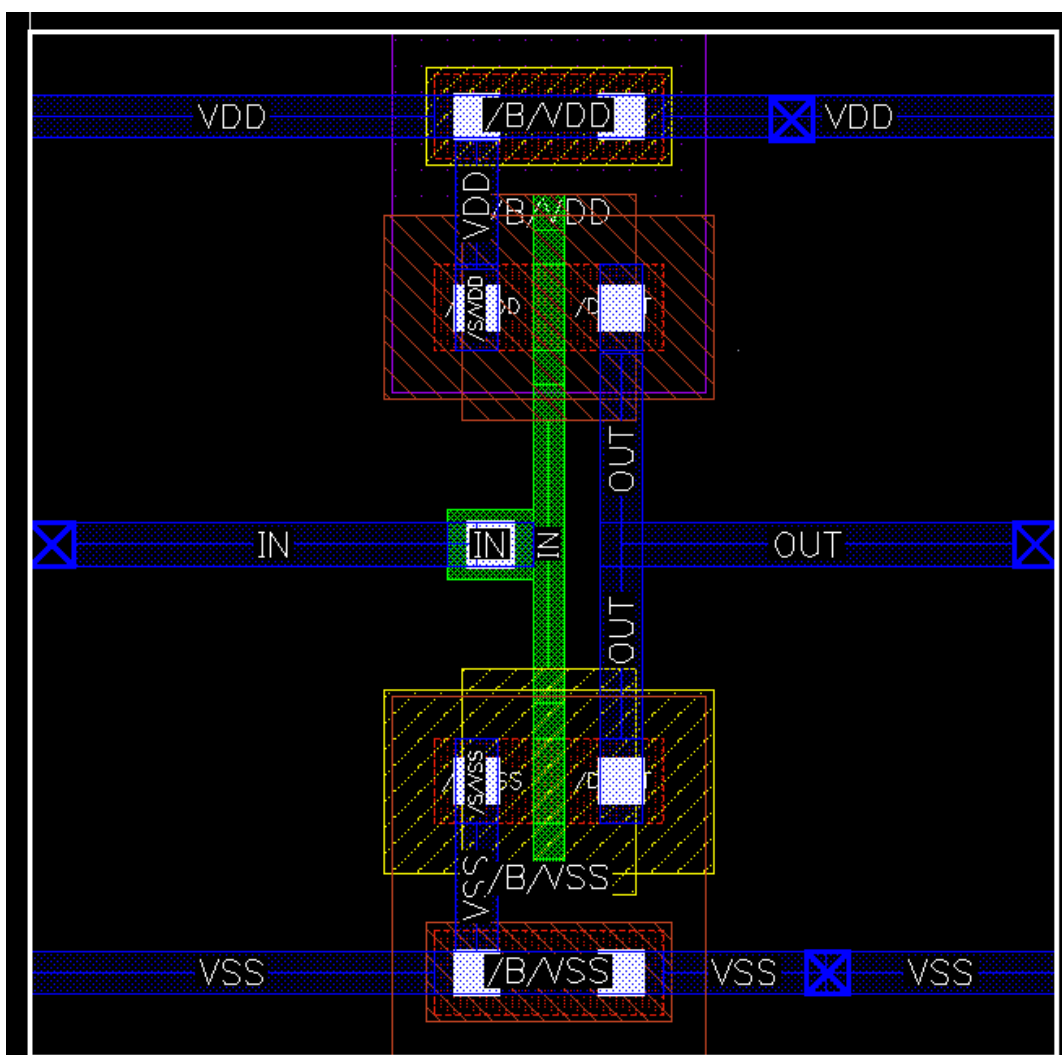
- 1 Откройте **Tools > Library Manager**.
- 2 Откройте schematic-представление ячейки INV и из неё создайте layout-представление **Launch > Layout L**. В открывшемся окне везде выберите *New File*. Далее выберите свою библиотеку, свой логический элемент INV, а также укажите тип *Type = layout*. В окне создания физического конфигурационного файла нажмите **Cancel**.
- 3 Перед Вами откроются будут открыты два окна — schematic и layout. Действия в одном окне будут коррелировать с действиями в другом (например, подсветка выделенных пино, проводников и транзисторов).
- 4 Откройте **Options > Display** и в открывшемся окне в блоке Grid Control установите значения X Snap Spacing и Y Snap Spacing равными 0.01, чтобы перемещение курсора по области редактирования топологии было более плавным. Нажмите ОК.
- 5 В schematic-представлении в свойствах транзистора (выделите транзистор и нажмите Q) в поле *Bodytie Type* укажите *detached*, тем самым контакт к подложке (Body) отделяется от транзистора. Ниже укажите месторасположение контакта к подложке: для р-канального транзистора — Top Tap, для n-канального — Bottom Tap.
- 6 Чтобы «подтянуть» layout-представления всех элементов из schematic-представления нажмите **Connectivity > Generate > All From Source**. В случае необходимости добавить только определённый элемент, то выберите его на schematic, а в layout нажмите **Connectivity > Generate > Selected From Source**.
- 7 В открывшемся окне просто нажмите ОК, на рабочей области появятся транзисторы, пины IN, OUT, VDD и GND, а также будет обозначена граница логической ячейки (boundary). Сочетанием клавиш Shift+F можно перевести условное изображение экземпляра элемента (instance) в его топологическое представление. Сочетанием Ctrl+F можно вернуть изображение экземпляра элемента (instance) в условное. Клавишей F можно уместить всю топологию под размер экрана.

- 8 Если всё верно, то при перемещении транзисторов, а также на их контактах будут отображаться линии, с какими контактами они связаны их названия соответственно.
- 9 Обратите внимание, что координата левого нижнего угла границы ячейки соответствует (0,0) — это **обязательное условие!** Эта граница определяет границу ячейки на этапе размещения и трассировки (Place & Route, P&R). Важным для будущих ячеек является необходимость в одинаковой высоте этой области!



- 10 Разместите транзисторы так, чтобы их затворы оказались на одной линии.
- 11 Для рисования проводников сначала выберите в списке слоёв нужный, а затем инструментом **Create > Wiring > Wire** (гор. клавиша P) нарисуйте проводники из Metal1 и Poly. Для этого наведите курсор на край проводника, которых хотите продолжить, ширина автоматически изменится на текущую, а слой станет слоем проводника, и проведите до края другого проводника.

12 Слой Metal1 соедините контакт подложки к контакту истока у обоих транзисторов.



13 Пин, который отвечает за подключение питания VDD (можно посмотреть в его свойствах на вкладке Attribute его имя, слой, которым он является, а на вкладке Connectivity — входной или выходной пин) переместите на соединение тела и истока транзистора. Или выведите отдельно дорожку металлизации к нему.

14 Аналогично проделайте с пином GND.

15 Соедините металлизацией Metal1 стоки транзисторов и выведите дорожку к пину OUT.

16 Выделите все пины, откройте свойства (клавиша Q) и на вкладке Attribute в поле Layer поменяйте подключение с Metal1 тип drawing на Metal1 тип pin.

17 Слой поликремния Poly соедините затворы транзисторов.

18 Для подключения затвора к пину IN, который представляет собой слой Metal1 используйте элемент Via. Сквозное отверстие (Via) создаётся инструментом **Create > Via** (гор. клавиша O). В рамках PDK достаточно указать в поле Via Definition, между какими слоями будет создано отверстие (поликремний и первый уровень металлизации M1_PO). Обратите внимание, что сквозные отверстия можно создавать только между **соседними** слоями металлизации!



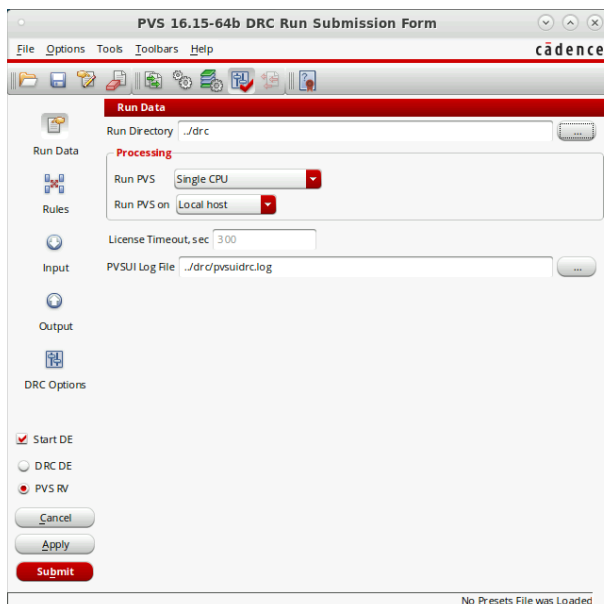
19 Используя **Create > Label** (гор. клавиша L) добавьте на соответствующие проводники метки IN, OUT, VDD, GND.

20 Создание топологии ячейки завершено. Сохраните ячейку.

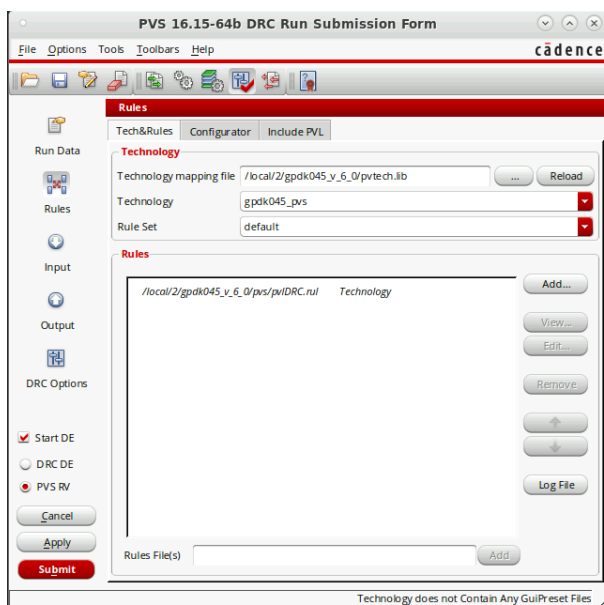
Проверка правил проектирования (DRC)

После создания топологии ячейки следует этап проверки выполнения правил проектирования (Design Rule Check, DRC).

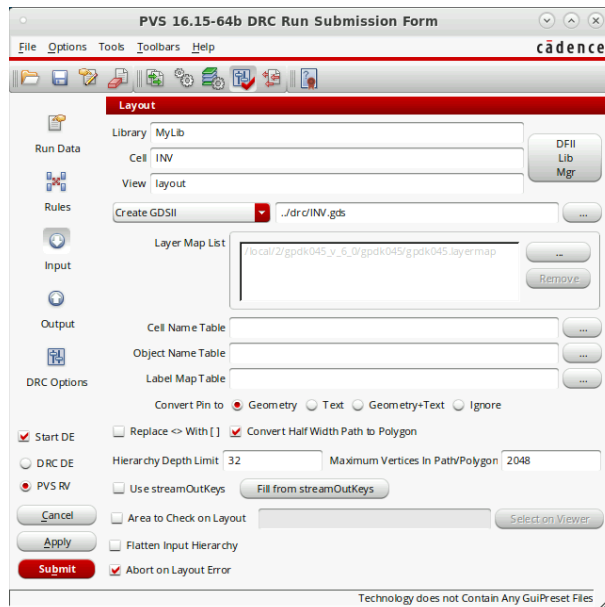
- 1 Для проведения DRC выберите меню **PVS > Run DRC**.
- 2 В блоке Run Data укажите директорию, в которую будут сохраняться результаты DRC [*напка_проекта*]/DRC.



3 В блоке Rules укажите папку с технологическим файлом *pvtech.lib*, расположенного либо в директории */local/1/gpdk045_v_6_0/*, либо в */local/2/gpdk045_v_6_0/*, технологию выберите Technology = *gpdk045_pvs*, набор правил укажите Rule Set = *default*. В списке ниже должен автоматически подгрузиться файл *pvsDRC.rul*, в котором прописаны правила проектирования.



4 В блоке Inputs можно выбрать, какую ячейку будет проверять анализ.



- 5 После нажать либо **Apply**, либо **Submit**. Начнётся проверка топологии на соответствие правилам проектирования. По окончании проверки появится окно с ошибками, либо проверка пройдёт удачно.
- 6 Если были ошибки, то следует их устранить, изменяя расположение элементов.
 - 6.1 Часто ошибки связаны с несоблюдением минимальных расстояний между соседними слоями или размеров этих слоёв. Для проверки удобно использовать линейку **Tools > Create Measurement** (гор. клавиша K).
 - 6.2 Также частыми ошибками является расположение элементов таким образом, что их координаты выходят за рамки сетки. В таком случае их также следует подвинуть, чтобы координаты углов не имели большего количества знаков после запятой, чем шаг сетки.
 - 6.3 Если ошибки возникают в геометрии экземпляра элемента (например транзистора) , то для получения доступа к её редактированию выделите элемент (instance) и используйте **Edit > Hierarchy > Flatten...** В открывшемся окне отметьте *Pcells* и в блоке *Preserve Pins*. Снять галочку *Vias*.
- 7 После исправления необходимо повторить проверку DRC нажатием кнопки ReRUN до того, как ошибки все исчезнут.

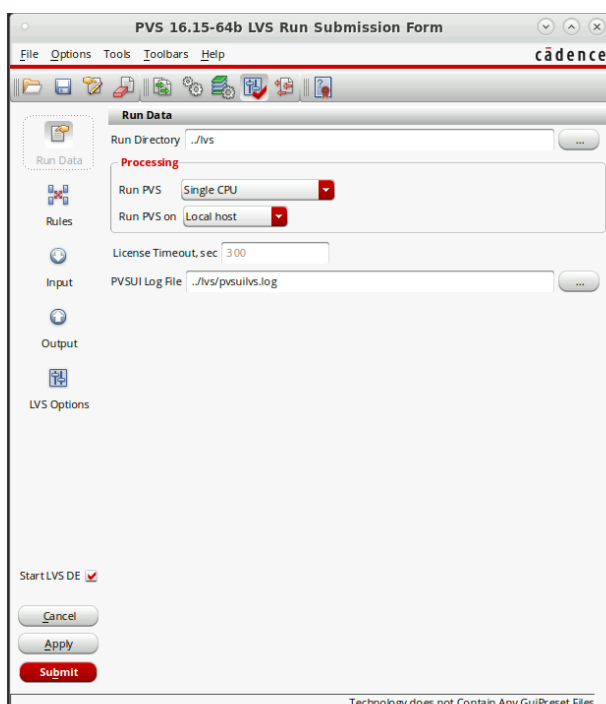
Проверка соответствия топологии и схмотехнического представлений (LVS)

Однако соблюдение правил проектирования не гарантирует, что топологическое изображение базовой ячейки соответствует её схмотехническому представлению. При рисовании топологии можно забыть

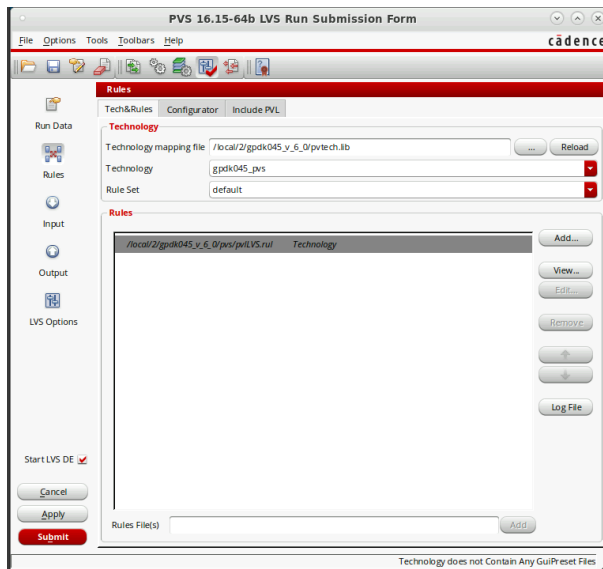
соединить какой-нибудь контакт, не довести дорожку металлизации, создать случайно несколько лишних пинов, наименовать их не в соответствии с именами в schematic.

Для проверки соответствия соединений в топологии со схемотехническим отображением проводится ещё один этап верификации топологии — LVS анализ (Layout vs. Schematic).

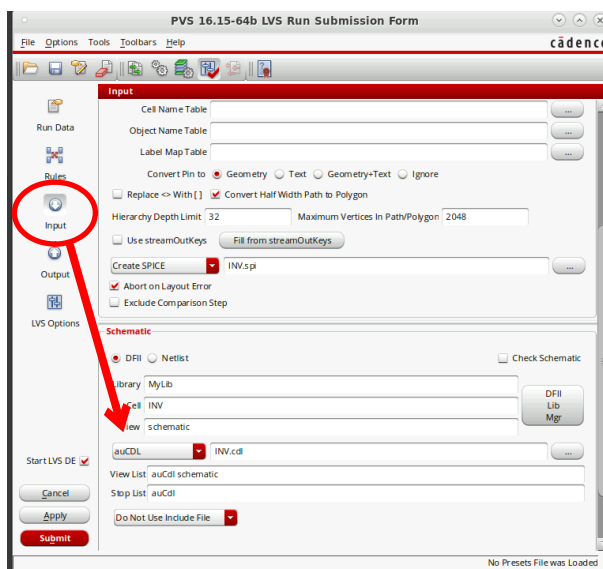
- 1 Откройте топологическое представление схемы.
- 2 Запустите **PVS > Run LVS**. Открывшееся окно похоже на окно с DRC.
- 3 По аналогии с DRC укажите в блоке Run Data директорию, в которую PVS будет сохранять файлы [*папка_проекта*]/LVS, затем загрузите файл с правилами для LVS pvsLVS.rul.



- 4 В блоке Rules на вкладке Tech&Rules укажите файл сопоставления технологии *pvtech.lib*, расположенный в директории с GPDК045. Выберите технологию *gpdk045_pvs*, а набор правил укажите *default*. В списке ниже должен автоматически подгрузиться файл *pvsLVS.rul*, в котором прописаны правила сопоставления топологии и электрической схемы.



- 5 В блоке Input убедитесь, что сравниваться будут два представления одной ячейки: schematic и layout (в области Layout и Schematic наименование библиотеки и ячейки должны совпадать).
- 6 Здесь же в области Schematic вместо Create CDL выберите auCDL.



- 7 В блоке Output внизу выставьте галочку в Additional Output на Create Quantus QRC Input Data. Будет указана папка /svdb, которая лежит в папке /lvs.
- 8 Запустите анализ (Apply или Submit).
- 9 Если были ошибки, то следует их устранить вместе с преподавателем. Подробное описание ошибок находится в папке [папка_проекта]/LVS в файле с расширением [имя_ячейки].rep.cls
- 10 Добейтесь отсутствия ошибок.

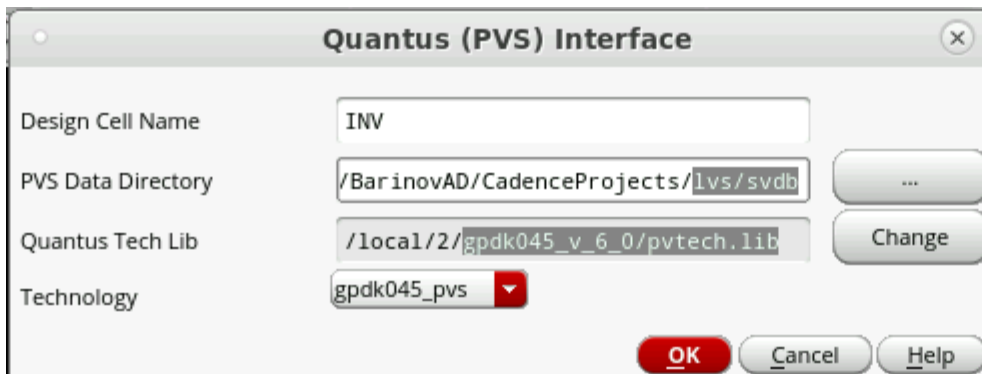


- 11 При необходимости использовать SPICE-описание схемы, восстановленного из топологии, можете его найти в папке [папка_проекта]/LVS в файле [имя_ячейки].spi

Экстракция паразитных параметров схемы (av_extracted-представление)

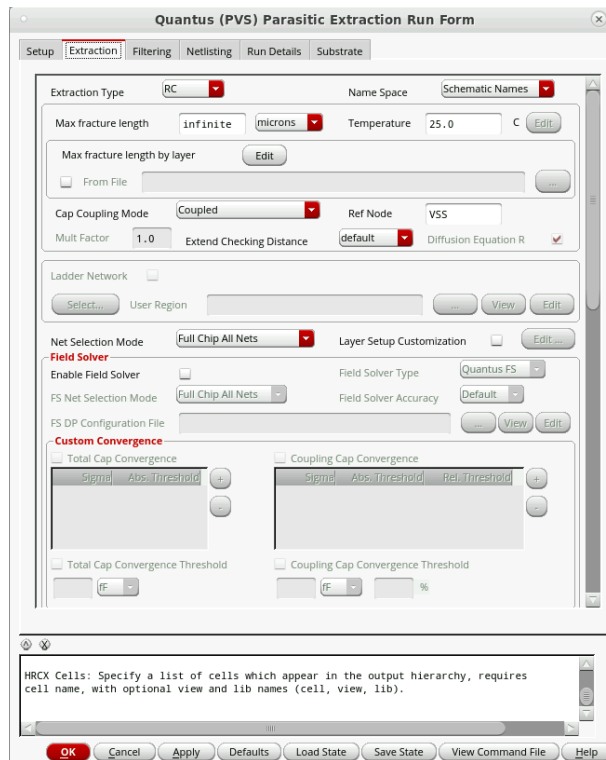
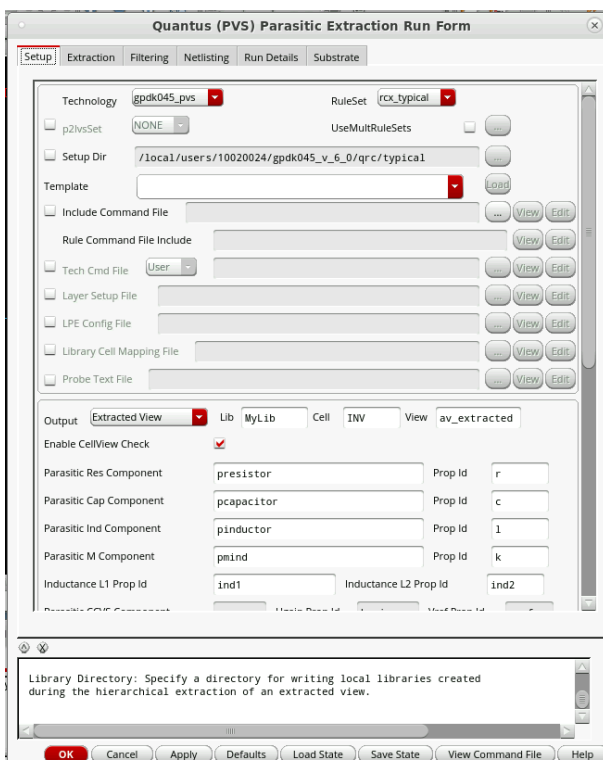
К сожалению, не смотря на удобство схемотехнического представления, данный уровень абстракции ограничивает учёт многих физических эффектов. На уровне топологии всегда есть дополнительные паразитные элементы — ёмкости металлических межсоединений и их сопротивления, индуктивности. Для получения более полного описания схемы используется операция экстракции паразитных параметров. Для экстракции используется модуль Quantus, который погружается в модуле EXT.

- 1 Откройте топологическое представление схемы.
- 2 Выберите **Quantus > Run PVS - Quantus**.
- 3 В открывшемся окне убедитесь, что указано верное имя ячейки, паразитные параметры которой экстрагируются.
- 4 В поле PVS Data Directory укажите путь к папке /lvs/svdb.
- 5 В поле Quantus Tech Lib укажите путь к папке с технологической библиотекой /gpdk045_v_6_0/pvtech.lib (возможно, необходимо будет переместить папку gpdk045_v_6_0 в домашнюю папку из-за проблем с доступом к файлу extview.rul).
- 6 Ниже выберите технологию gpdk045_pvs.



7 Нажмите ОК.

8 Перед Вами окно с настройками Quantus. Укажите параметры на вкладках Setup и Extraction такими, как показано на рисунках ниже.

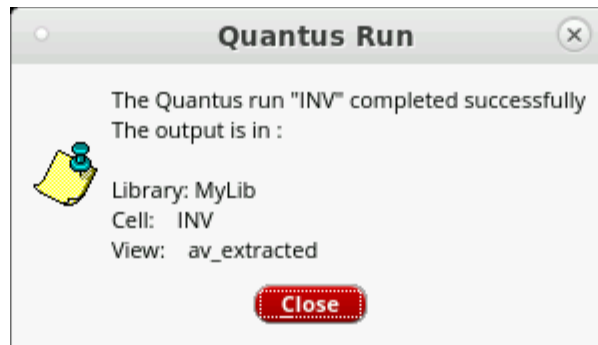


9 Нажмите ОК.

10 Начнётся процесс экстракции. В зависимости от сложности схемы длительность данного процесса может быть высокой.



- 11 В случае удачного завершения процесса появится сообщение о завершении.



- 12 Процесс экстракции паразитных параметров окончен и в библиотеке компонентов у ячейки INV появилось ещё одно представление - av_extracted.
- 13 Для получения SPICE-описания схемы с экстрагированными параметрами необходимо повторить пп. 1-7, а в п. 8 на вкладке Setup выбрать в строчке *Output* SPICE. В этом случае после экстракции паразитных параметров в папке /cds появится файл INV.sp.

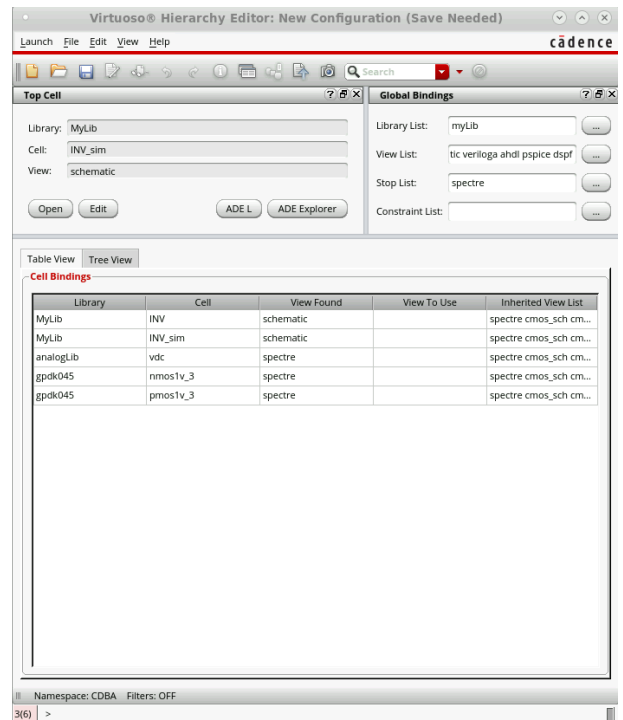
Создание конфигурационного файла

Теперь ячейка описана в четырёх представлениях - схемотехническом, символическом, топологическом и экстрагированном. Для того, чтобы её можно было моделировать с использованием разных описаний, например, для сравнения результатов моделирования идеальной схемы без паразитных параметров и реальной схемы с паразитными параметрами, следует создать конфигурационный файл. Таким образом, конфигурационный файл позволит моделировать схему как в schematic-представлении, так и в layout-представлении.

Создание config-представления

- 1 В Library Manager откройте schematic-представлении схемы моделирования инвертора INV_sim.
- 2 Выберите **File > New...** и в появившемся окне укажите то же имя ячейки моделирования, а тип выберите config. Нажмите ОК.
- 3 Появится окно настройки конфигурации.
- 4 В настройке Top Cell выберите вид View = schematic.
- 5 Нажмите внизу кнопку Use Template и выберите симулятор spectre. Нажмите ОК.
- 6 Нажмите ОК.

7 Сохраните файл конфигурации.



8 Закройте окно конфигурации.

9 Закройте окно symbol-представления моделирования инвертора.

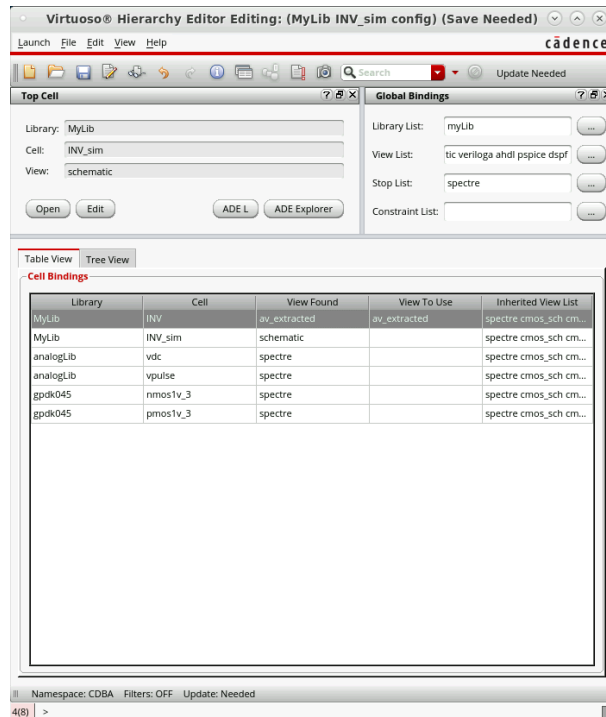
Моделирование нескольких представлений (schematic и av_extracted)

- 1 Теперь необходимо открыть конфигурационный файл. В Library Manager выберите ячейку INV_sim и откройте её config-представление.
- 2 В открывшемся окне везде укажите yes. Нажмите ОК.
- 3 Откроются два окна: редактор иерархии Hierarchy Editor и schematic-представление схемы моделирования в режиме config.
- 4 Запустите моделирование **Launch > ADE L**.
- 5 Настройте анализ реакции инвертора на импульс при напряжении питания 1,1 В. Импульс от 0 до 1,1 В, период 2 нс, время нарастания и спада - 0,2 нс. Время анализа - 10 нс. Промоделируйте сигнал на входе и выходе инвертора.
- 6 Нажмите правой кнопкой мыши по выходной осциллограмме и выберите Trace Properties. В открывшемся окне измените наименование кривой, добавив окончание «-schem», например «/OUT-schem».
- 7 При необходимости можно изменить вид кривой осциллограммы, нажав правой кнопкой мыши по ней и выбрав Trace Properties в контекстном меню, либо выбрав в списке внизу этого контекстного

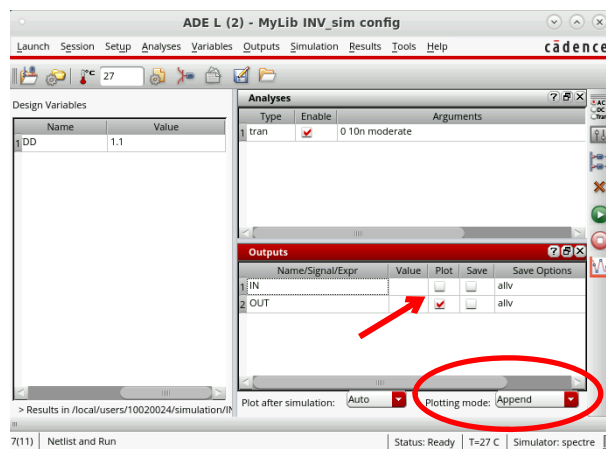
меню необходимые изменения: стиль, цвет и толщину линии осциллограммы.

8 Не закрывайте окно ADE L.

9 В окне Hierarchical Editor выберите свой инвертор INV, нажмите правой кнопкой в столбце View Found и в открывшемся контекстном меню выберите **Set Cell View > av_extracted**. Сохраните изменения. В Tree View добавится множество паразитных резисторов и емкостей.

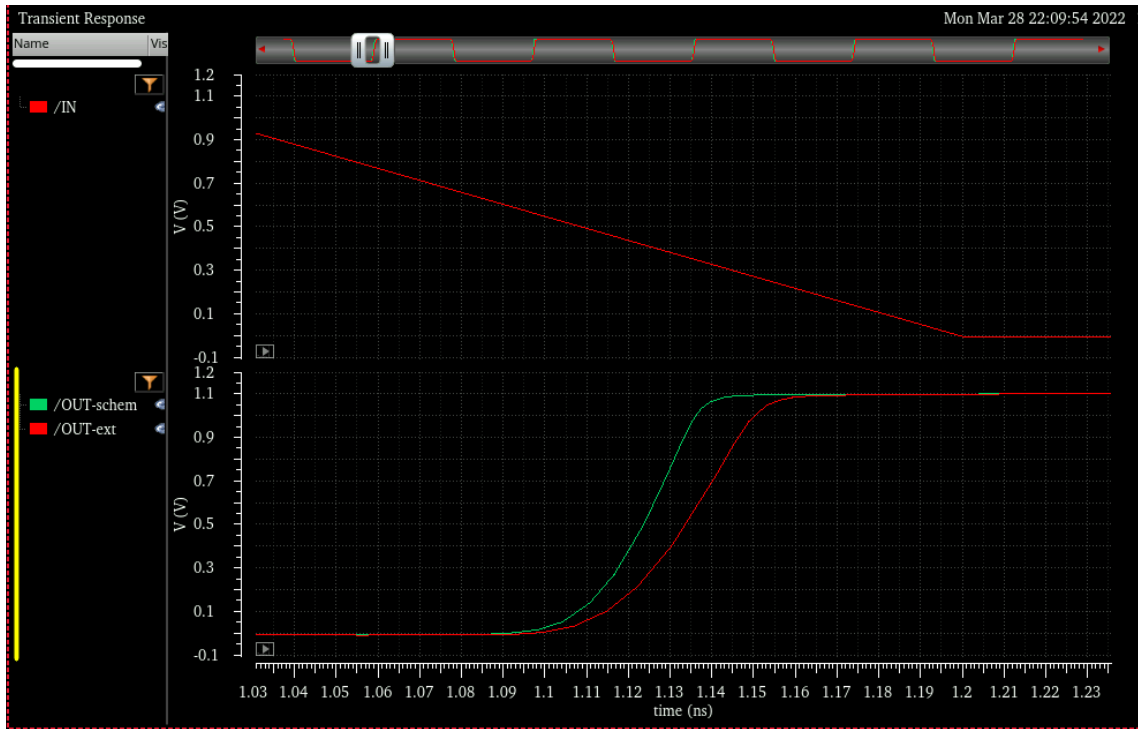


10 В окне ADE L снизу выберите представление нового результата моделирование Plotting mode не Replace, а Append. В списке выводимых сигналов снимите галочку *Plot* с сигнала IN. Запустите моделирование.



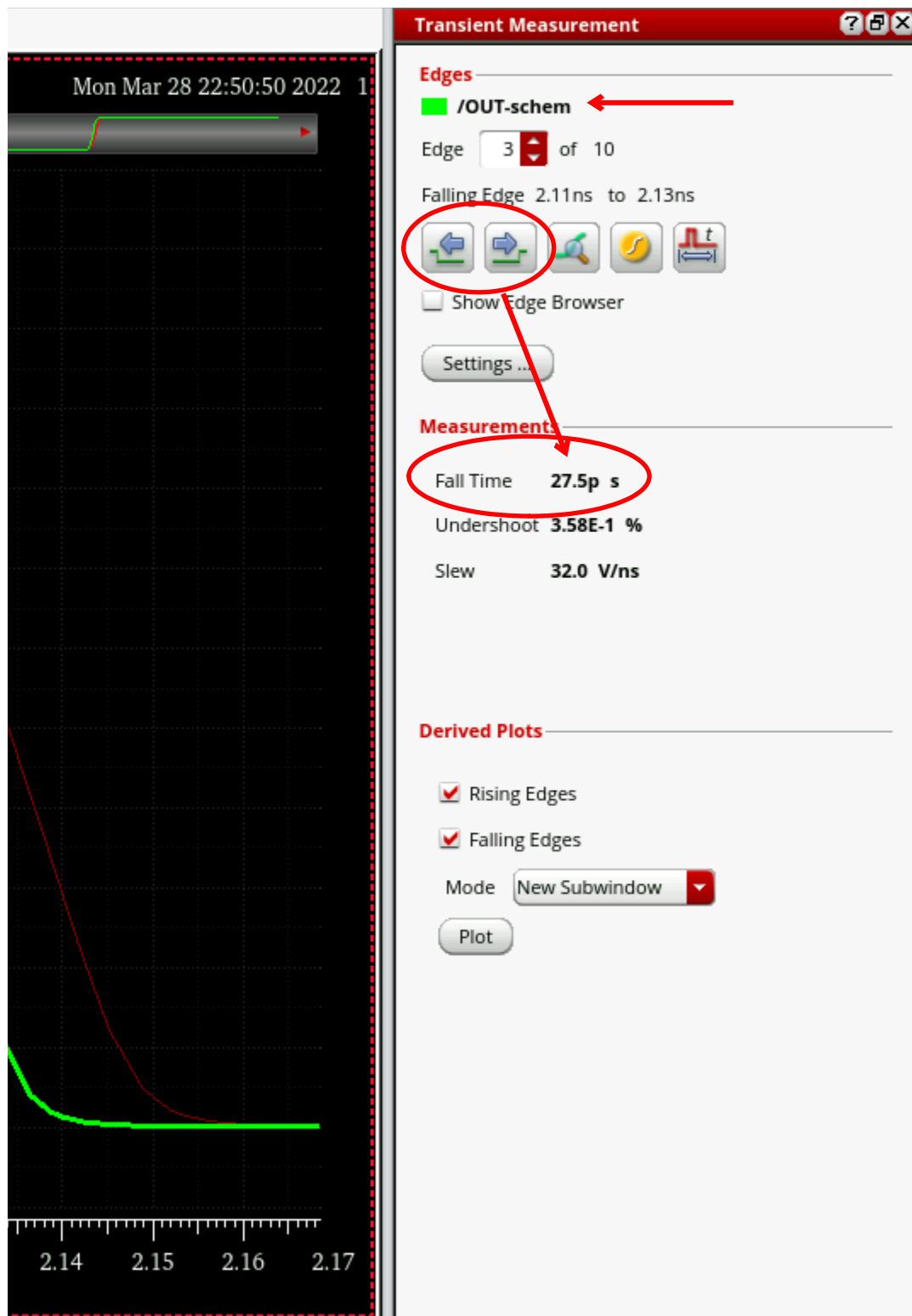
11 На графике результата моделирования добавится вторая кривая у OUT для схемы с экстрагированными паразитными параметрами. Переименуйте её, например, в /OUT-ext. В зависимости от реализации

топологии результаты между `av_extracted` и `schematic` могут сильно различаться. На графике ниже зелёная нижняя кривая – фронт переключения для `schematic`, красная – для экстрагированной схемы. Как видно, фронты различаются достаточно сильно.



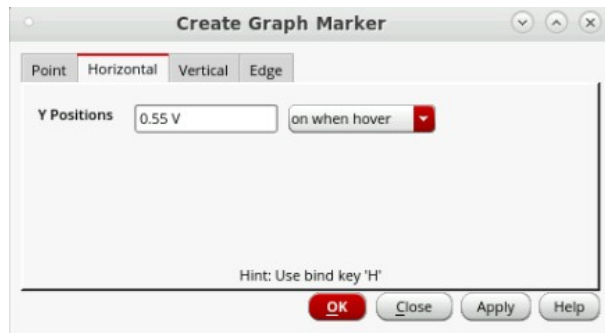
Измерение длительностей переднего и заднего фронтов

- 1 Выберите на графике кривую `/OUT-schem`, затем выберите меню **Measurments** > **Transient Measurment**. Справа от графиков появится окно для настройки и проведения измерений.
- 2 Выберите в меню справа **Settings**, в открывшемся окне укажите порог для измерений *Threshold* 10%-90%.
- 3 При помощи кнопок *Previous edge* и *Next edge* перемещайтесь по фронтам и в блоке **Measurment** будет появляться значение **Rise Time** или **Fall Time**. Запишите их.
- 4 Повторите пп. 1-3 для сигнала результата моделирования с экстрагированными паразитными параметрами `/OUT-ext`.

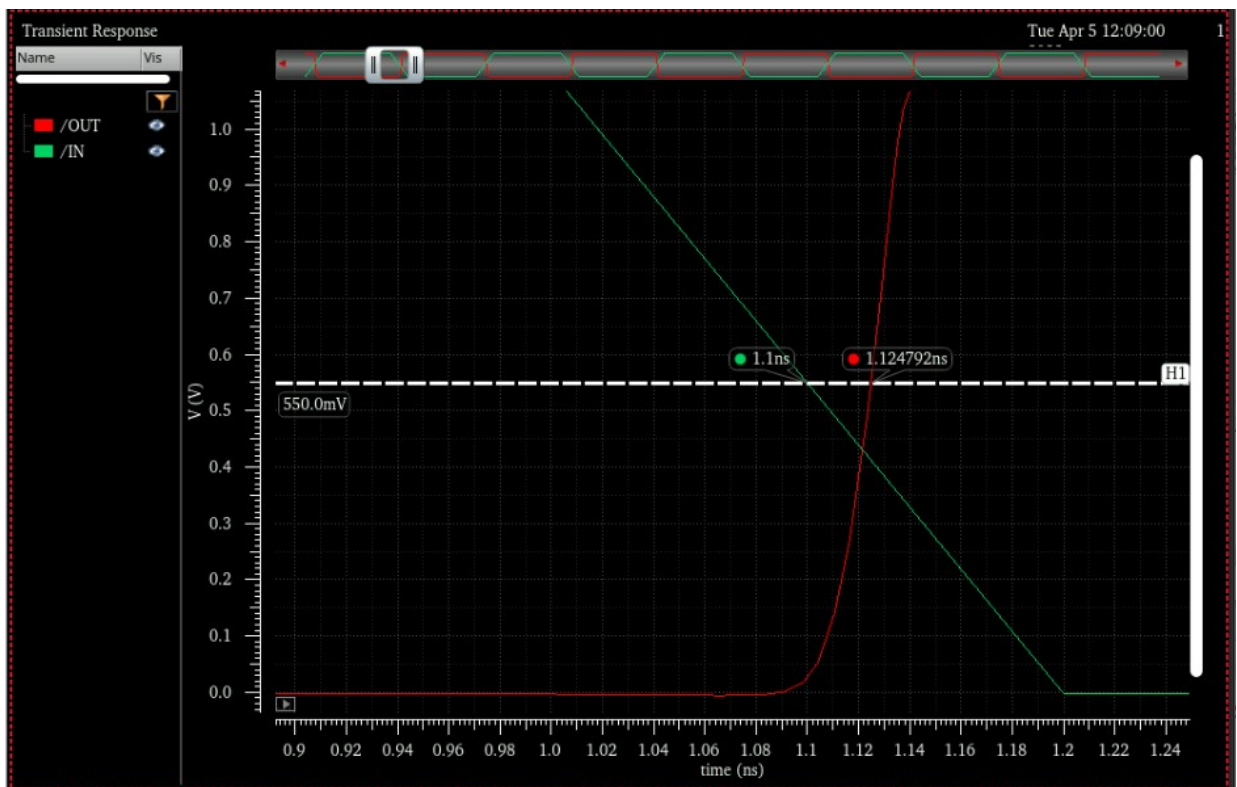


Измерение времени задержки распространения сигнала

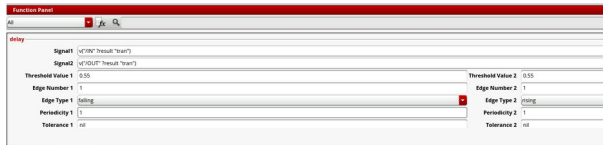
- 1 Совместите на одном графике все кривые (входной импульс и оба выходных импульса).
- 2 Выберите **Marker > Create Marker...** В открывшемся окне на вкладке *Horizontal* укажите положение маркера по оси напряжений равную половине напряжения питания (50 % от Vdd). Нажмите ОК.



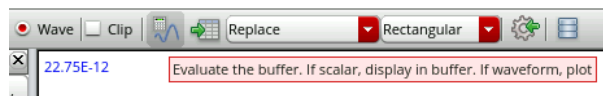
- 3 На графике появится горизонтальная линия-маркер, на пересечении с графиками входных и выходных сигналов при наведении на линию-маркер появятся значения времён, при которых достигается указанный уровень напряжения.



- 4 Время задержки распространения сигнала определяется как разница времён на уровне 50 % от Vdd. То есть для указанного выше рисунка время задержки распространения сигнала при переключении из 0 в 1 (красная кривая) составляет $1.124792 - 1.1 = 24,792$ пс.
- 5 Второй способ – расчёт при помощи встроенного калькулятора. Для этого используйте **Tools > Calculator**.
- 6 В открывшемся окне снизу в области Function Panel выберите All и найдите там delay.
- 7 В изменившемся окне укажите, что первый сигнал, что Вы используете – это IN, а второй – OUT.



- 8 Форма сигнала для IN – falling, для OUT – rising (измерение задержки распространения времени переключения из 0 в 1). В качестве порога напряжения укажите 50 % от Vdd (Threshold Value 1 и 2 = 0,55 В). Edge Type должны быть для текущей схемы противоположны (положительному фронту на входе инвертора соответствует спадающий фронт на выходе, для иных схем может различаться).
- 9 Далее используйте расчёт значения через буфер



- 10 Рассчитайте времена переключения из 0 в 1 и из 1 в 0.

Самостоятельная работа

- 1 Сформируйте топологию ячеек INV, NAND2 и NOR2. Для ячеек NAND2 и NOR2 расположите в единую горизонтальную линию р-канальные и n-канальные транзисторы таким образом, чтобы у них были соответственно общие диффузионные области.
- 2 Проведите для них DRC-анализ.
- 3 Проведите для них LVS-анализ.
- 4 Произведите экстракцию паразитных параметров для всех схем – создание как av_extracted варианта, так и SPICE-описания с паразитными параметрами.
- 5 Произведите моделирование реакции на импульс всех схем с расчётом времён переключения (Rise Time и Fall Time), а также времён задержек распространения сигнала (Time Delay) при переключении из 0 в 1 и из 1 в 0 как у schematic, так и у av_extracted. Параметры импульса возьмите из Практического занятия № 1.

Содержание отчёта

Отчёт должен содержать:

- 1 Титульный лист.
- 2 Задание.
- 3 Комментарии к выполнению.
- 4 Топологию инвертора.
- 5 Положительный результат DRC-анализа (показать преподавателю напрямую).
- 6 Положительный результат LVS-анализа (показать преподавателю напрямую).
- 7 ПП. 4—6 для ячеек NAND2 и NOR2.
- 8 SPICE-описание ячеек с экстрагированными паразитными параметрами
- 9 Электрическую схему, восстановленную по описанию п. 8, с пояснением того, что характеризует тот или иной паразитный элемент.
- 10 Результат моделирования реакции на импульс, параметры которого указаны в Практическом занятии № 1, для схем без паразитных параметров (schematic) и с паразитными параметрами (av_extracted). Результат представить в виде изображения фронтов.
- 11 Таблицу с результатами измерения длительности переходного процесса.

Схема	Время переключения из 0 в 1 (Time Rise), пс		Время переключения из 1 в 0 (Time Fall), пс	
	без паразитных элементов	с паразитными элементами	без паразитных элементов	с паразитными элементами
INV				
NAND2				
NOR2				

12 Таблицу с результатами измерения времён задержки распространения сигнала и быстродействием (определяется как среднее время переключения из 1 в 0 и из 0 в 1).

Схема	Время задержки распространения сигнала при переключении из 0 в 1 (Time Delay), пс		Время задержки распространения сигнала при переключении из 1 в 0 (Time Delay), пс		Быстродействие (Time Delay), пс	
	без паразитных элементов	с паразитными элементами	без паразитных элементов	с паразитными элементами	без паразитных элементов	с паразитными элементами
INV						
NAND2						
NOR2						

Практическое занятие № 3. Создание топологического изображения компонента

Самостоятельная работа

- 1 На основе предыдущих работ сформируйте схематехническое представление (schematic) компонента, заданного вариантом ниже.
- 2 Сформируйте его символьное представление (symbol).
- 3 Сформируйте его топологическое представление (layout) в среде Layout XL, используя в ней **Connectivity > Generate > All From Source...** Обратите внимание, что:
 - 3.a все иерархические блоки должны быть одной высоты;
 - 3.b у всех иерархических блоков должны быть соединены в одну линию металлизации VDD и GND (VSS), которые должны быть максимально удалены друг от друга;
 - 3.c рекомендуется для соединения между базовыми блоками использовать двухуровневую металлизацию (Metal1 для горизонтальных линий и Metal2 для вертикальных);
 - 3.d переход между уровнями металлизации осуществляется при помощи сквозного отверстия Via;
 - 3.e для скоростного перехода между уровнями при формировании ручной разводки используйте нажатие правой кнопки с выбором *Via Up to Metal2* для перехода с уровня Metal1 на Metal2 и *Via Down to Metal1* для перехода с уровня Metal2 на Metal1;
 - 3.f внутренняя металлизация не должна выходить за пределы шин питания и земли (за границы prBoundary компонента).
- 4 Выделите все пины, откройте свойства (клавиша Q) и на вкладке Attribute в поле Layer поменяйте подключение с *Metal1* тип *drawing* на *Metal1* тип *pin*.
- 5 Проведите анализы DRC, LVS.
- 6 Проведите экстракцию паразитных параметров (av_extracted).
- 7 Промоделируйте схему без наличия и с наличием паразитных параметров.

Содержание отчёта

- 1 Титульный лист.
- 2 Задание.
- 3 Комментарии к выполнению.

- 4 Схемотехническое и символическое изображения компонента схемы.
- 5 SPICE-описание схемы с паразитными параметрами.
- 6 Результаты моделирования схемы без и с паразитными параметрами.
- 7 Выводы о влиянии паразитных параметров на работу схемы.

Таблица 3.1 – Варианты схемы

Номер варианта	Схема
1, 11, 21	DFFSR (D-триггер с установкой и сбросом)
2, 12, 22	Мультиплексор «из 4 в 1»
3, 13, 23	Шифратор «из 4 в 2»
4, 14, 24	Дешифратор «из 2 в 4»
5, 15, 25	DFFS (D-триггер с установкой)
6, 16, 26	DFFR (D-триггер со сбросом)
7, 17, 27	Одноразрядный полный сумматор
8, 18, 28	Одноразрядный полусумматор
9, 19, 29	Демультимплексор «из 1 в 4»
10, 20, 30	DFF (D-триггер без установки и сброса)

Практическое занятие № 4. Генерация LEF-файла

LEF (Library Exchange Format, формат обмена библиотеками) — это файл, который абстрактно описывает расположение стандартных ячеек и используется для инструментов автоматического размещения и маршрутизации (P&R). Файл LEF имеет читаемый формат ASCII, включая подробную информацию о выводах для межсоединений, но не включает низкоуровневые слои, такие как поликремний и диффузия и пр.

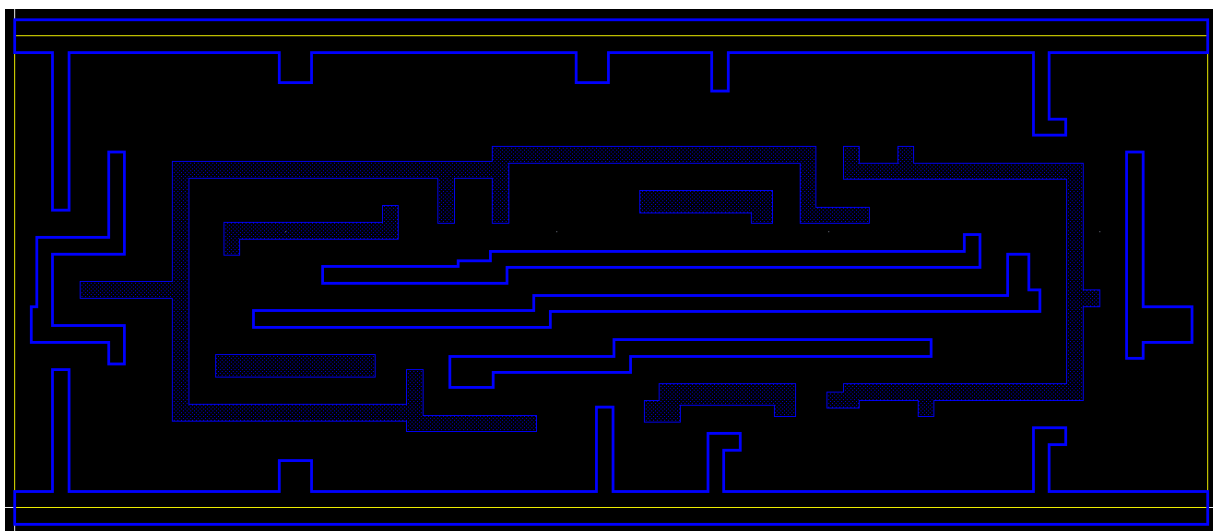


Рисунок 4.1 — Пример абстрактного представления

Файл LEF состоит из двух частей:

1. Часть с описанием технологического процесса (инструмент Virtuoso)
 - 1) Слои, доступные в технологическом процессе (Layer), включают только слои, используемые в процессе P&R.
 - 2) Некоторые правила проектирования, влияющие на работу P&R, такие как минимальная ширина дорожки металлизации и расстояние между ними.
 - 3) Правила маршрутизации, определенные разработчиком библиотеки, такие как выбранное значение минимального расстояния между дорожками металлизации, предпочтительная ориентация маршрутизации металла или геометрическое описание используемых переходных отверстий («переходные отверстия», via, здесь также включают два слоя металла, соединенных со слоем разреза), требующих уширения дорожки металлизации).

- 4) (опционально) Электрические свойства слоев в библиотеке, такие как максимальный ток, поверхностное сопротивление используемых слоев металлизации.

```
LAYER Metal1
TYPE ROUTING ;
DIRECTION HORIZONTAL ;
PITCH 0.6 0.6 ;
WIDTH 0.3 ;
SPACING 0.3 ;
RESISTANCE RPERSQ 0.101 ;
CAPACITANCE CPERSQDIST 0.000132 ;
EDGECAPACITANCE 8.8e-05 ;
MINIMUMDENSITY 0.25 ;
ANTENNAMODEL OXIDE1 ;
    ANTENNAAREARATIO 200 ;
DCCURRENTDENSITY AVERAGE 2 ;
END Metal1

LAYER Via1
TYPE CUT ;
SPACING 0.3 ;
WIDTH 0.2 ;
ANTENNAMODEL OXIDE1 ;
    ANTENNAAREARATIO 20 ;
DCCURRENTDENSITY AVERAGE 0.1 ;
END Via1
```

CSDN @淇则有洋

Рисунок 4.2 — Пример описания слоя металлизации Metal1, а также слоя сквозного отверстия Via1

2. Часть с описанием ячеек (инструмент Abstract)

- 1) Форма и размер ячеек, определяемые их соответствующими границами.
- 2) Расположение пинов, слои, на которых расположены пины, и геометрическое описание других фигур в том же узле.
- 3) Подробное описание барьеров, то есть фигур в слое межсоединений, которая не принадлежит какому-либо конкретному контакту, но запрещают прохождение дорожек металлизации в том же слое через них.

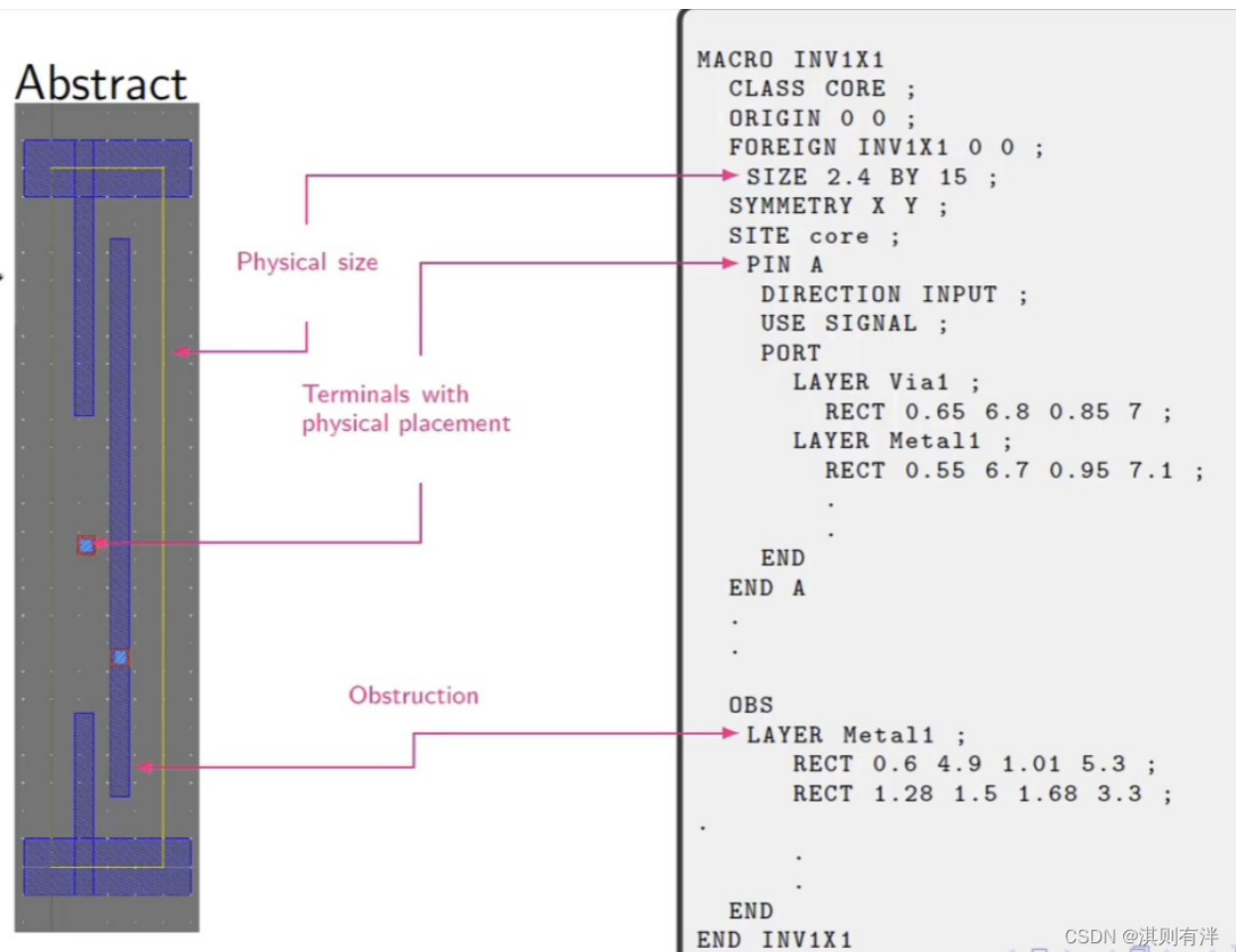


Рисунок 4.3 — Пример абстрактного представления ячейки инвертора и её lef-описание

Следует отметить, что технологическая часть lef-файла и часть с описанием ячеек могут быть расположены в отдельных lef-файлах: технологическом (tech lef) и со стандартными ячейками (stdcells lef).

Самостоятельная работа

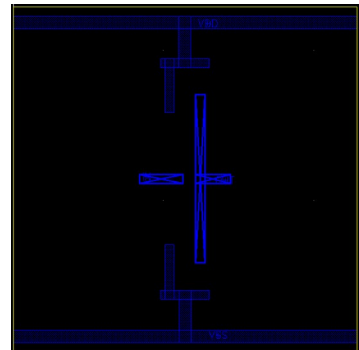
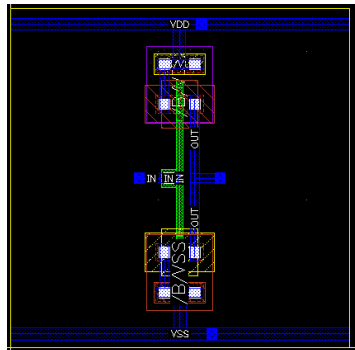
Сформируйте lef-файл для всех ячеек в библиотеке (стандартные вентили, компонент интегральной схемы и стандартной ячейки с заданной логической функцией).

Использование инструмента virtuoso

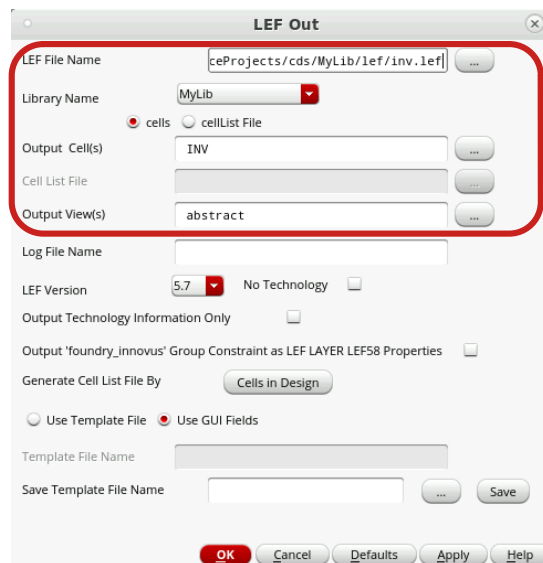
1. Откройте layout-представление ячейки, для которой необходимо сгенерировать lef-файл.
2. Откройте меню **Tool > Abstract Generation...** Нажмите ОК в всплывающем окне. Выберите библиотеку, в которой располагается ячейка, выберите ячейку. Укажите то layout-представление, из которого будет формироваться абстрактное представление. Отметьте тип ячейки

(в нашем случае это Core), укажите верхний уровень металлизации (обычно это Metal1). Нажмите Load Options, а затем Generate Abstract.

3. В библиотеке должно появиться абстрактное представление ячейки. Ниже представлено layout- и abstract-представления инвертора.



4. В CIW-окне Virtuoso выберите **File > Export > LEF...**. В открывшемся окне укажите имя создаваемого lef-файла (сохраните файл в папке /lef в папке проекта или библиотеки). Укажите библиотеку, к которой этот файл принадлежит. Укажите все ячейки, которые должны быть в данном lef-файле (для этого заранее необходимо сформировать их абстрактные представления). Нажмите ОК.

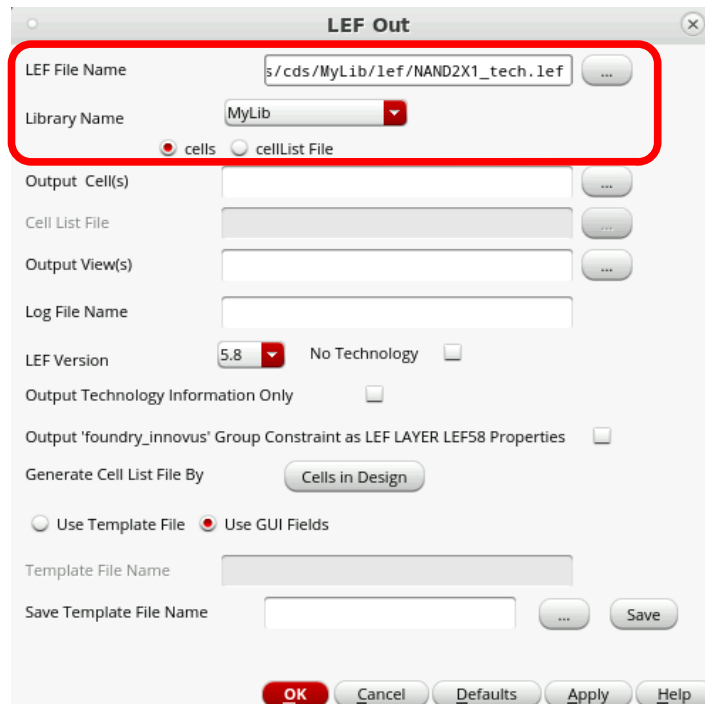


5. Закройте virtuoso.

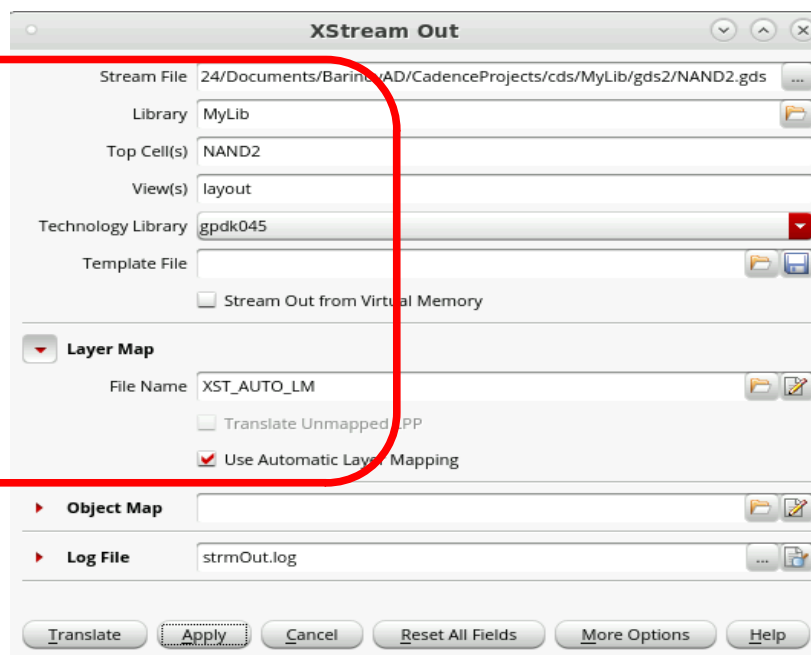
Использование инструмента abstract

1. Для формирования описания технологической части откройте Virtuoso, в окне CIW выберите **File > Export > LEF**. В открывшемся окне введите название будущего lef-файла (при необходимости создайте папку, в которую будете сохранять файл), в поле Library Name

выберите библиотеку с ячейками, которая будет экспортироваться и нажмите ОК.

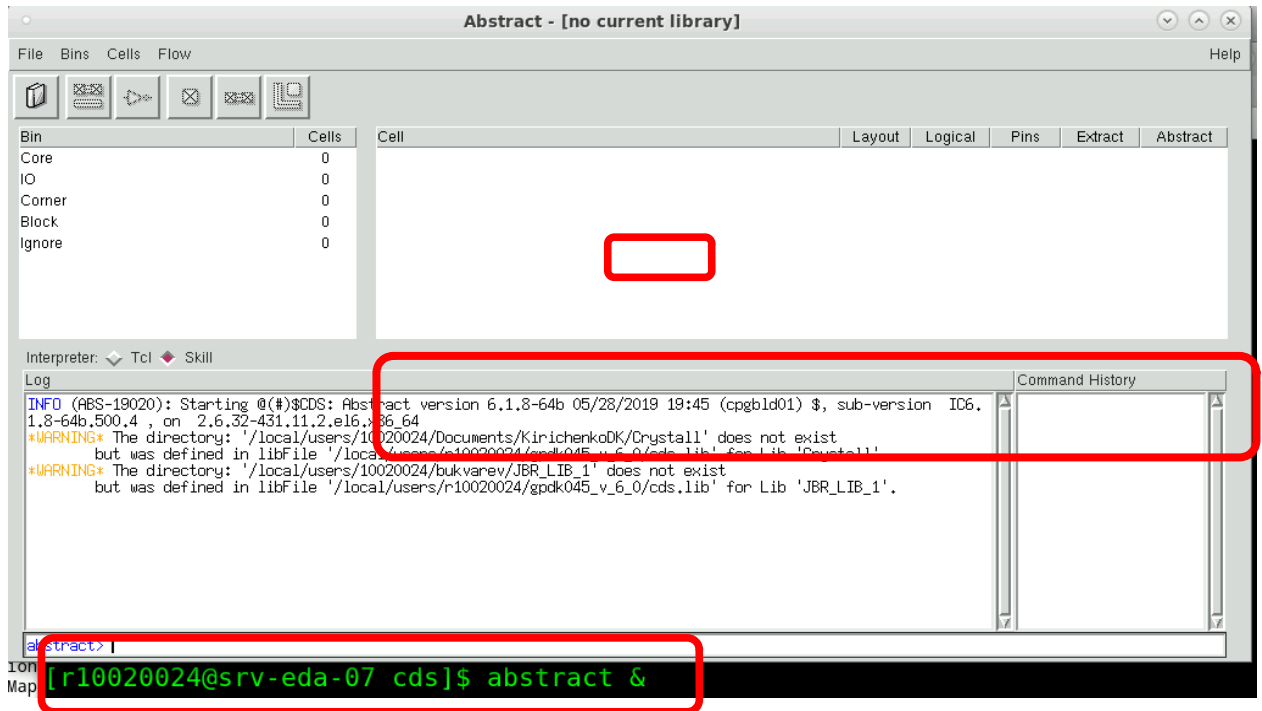


2. Для экспорта GDS-файла выберите в окне CIW **File > Export > Stream**, введите путь для экспорта и название экспортируемого файла. Выберите файл со слоями, который должен быть экспортирован (выберите **Library > Cell > layout** представления). В поле Layer Map укажите автоматическую генерацию соответствия слоёв. Нажмите Translate.

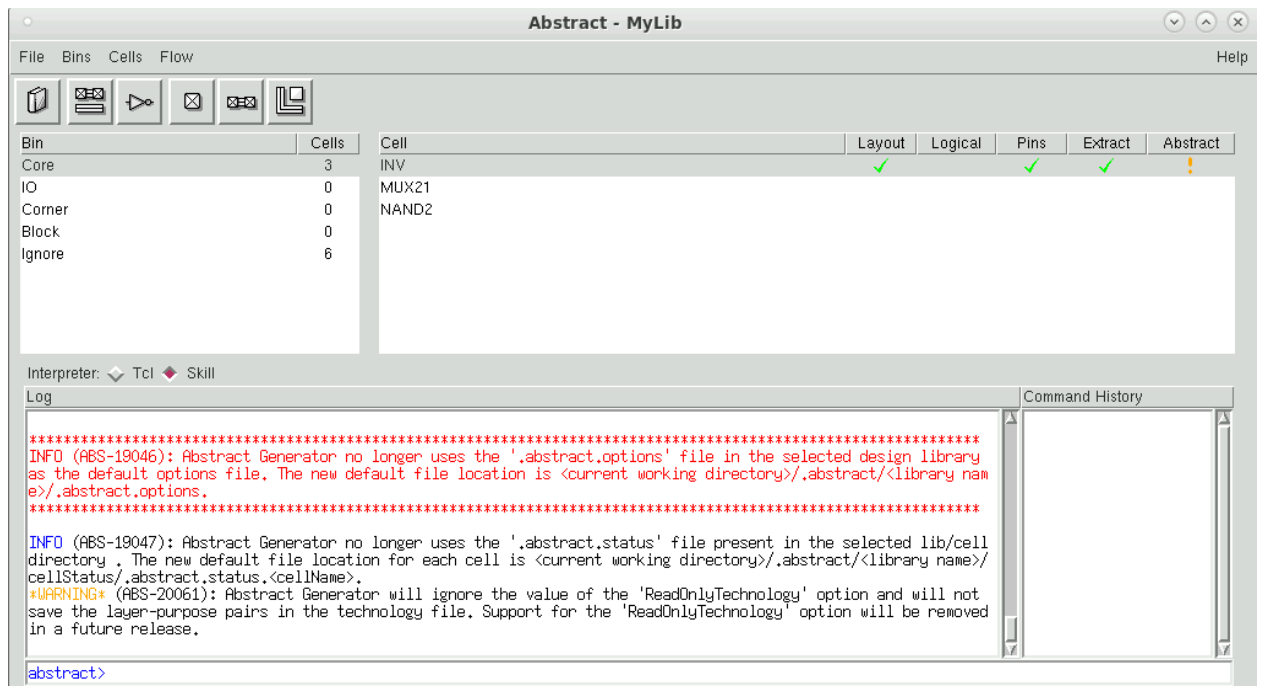


3. Откройте генератор абстрактного представления в папке /cds

\$ abstract &



4. Затем в открывшемся окне выберите **File > Library > Open...**, выберите свою библиотеку. В окне Abstract появится список ячеек библиотеки с отметкой того, какие пункты создания



5. Выберите из списка ячеек ту, для которой необходимо сформировать абстрактное представление и выберите **Flow > Pins....** Здесь мы выбираем параметры для извлечения форм пинов из меток в layout-представлении и направленности пинов в логическом представлении. Заполните в поле Clock имя пина для тактового сигнала, в поле Outputs

наименование выходных пинов в столбце Mar. Также укажите слой для отображения подписей пинов.

6. Выберите столбец Boundary и отметьте, если необходимо внести какие-либо коррективы в положение границы ячейки. Например, если необходимо, чтобы граница была на 0,4 мкм выше верхней и ниже нижней части пинов питания (или верхней / нижней границы металлизации), введите значение 0,4 в текстовые поля ввода сверху и снизу в поле Adjust Boundary By.
7. Нажмите кнопку Run и дождитесь завершения этапа генерации пинов. Выдается предупреждение, когда часть ячейки выходит за пределы (из-за того, что часть n- или p-кармана находится за пределами границ). Их можно игнорировать, поскольку на этапе размещения они будут примыкать к карманам одинаковой полярности в соседних ячейках.
8. Нажмите значок Extract и щёлкните Run без модификации.
9. Нажмите значок Abstract, выровняйте столбец, установите ширины, смещения и формы направляющих питания (измерьте их в layout-представлении).
10. В столбце Blockage удалите неиспользуемые слои металлизации и сквозных отверстий.
11. В поле site (месторасположение) укажите название месторасположения, например «core».
12. Столбец Grid позволяет увидеть, являются ли данные назначения неправильными.
13. Нажмите Run для запуска.
14. Нажмите на значок Verify, и команда вызовет Silicon Ensemble и построит тестовую схему для проверки маршрутизации файла LEF, созданного этими устройствами. Это предупредит о потенциальных проблемах. Нажмите Run для запуска.
15. Наконец выберите **File > Export > LEF** и экспортируйте сгенерированный LEF-файл.
16. Откройте файл LEF, сгенерированный Abstract, и выберите для копирования из инструкции определения первого MACRO в инструкцию END последнего MACRO.

17. Откройте файл LEF, экспортированный Virtuoso, и вставьте то, что вы только что скопировали, между инструкцией END и инструкцией END LIBRARY последнего слоя.

18. Укажите информацию о SITE между инструкцией MANUFACTURINGGRID и инструкцией Layer в следующем формате:

```
SITE siteName
```

```
CLASS {PAD | CORE}
```

```
SYMMETRY {X | Y | R90}
```

```
SIZE width BY height
```

```
END siteName
```

19. Сохраните документ.

Содержание отчёта

- 1 Титульный лист.
- 2 Задание.
- 3 Комментарии к выполнению.
- 4 layout- и abstract-представление ячеек и компонента схемы.
- 5 LEF-файл (или его содержание).

Практическое занятие № 5. Генерация LIB-файла

Файлы формата liberty (lib-файлы) содержат параметры, используемые при характеристике стандартных ячеек. К параметрам характеристики относятся два класса, применяемых для разного рода моделирований:

1. Для функционального моделирования:
 - 1) описание функциональной работы ячейки.
2. Для оптимизации по какому-либо параметру, для моделирования времён задержек:
 - 1) информация о площади ячейки;
 - 2) информация о потребляемой мощности;
 - 3) временные ограничения (setup / hold времена, recovery / removal времена);
 - 4) информация о времени задержки распространения сигнала (propagation delay).
3. Для извлечения информации о мощности, анализа падений напряжения, анализа электромиграции:
 - 1) информация о динамической мощности, потребляемой в момент переключения транзисторов;
 - 2) информация о статической мощности, определяемой токами утечки;
 - 3) информацию о пассивной мощности, являющейся внутренней мощностью, проявляемой в момент, когда входные сигналы последовательностных ячеек (таких как D-триггеры) изменяются, но при этом выходной сигнал ячеек остаётся неизменным.
4. Для размещения и трассировки:
 - 1) информация о геометрии ячеек (ширина ячейки);
 - 2) информация о расположении пинов;
 - 3) информация о каналах для трассировки (области металла, неиспользуемые ячейкой) и преград для трассировки (дополнительные области металла, используемые ячейкой).

Линейная модель задержек предполагает использование RC -цепочки, где R — сопротивление открытого канала транзисторов, C — суммарная ёмкость нагрузки.

Данная модель имеет следующие проблемы. Нагрузочный конденсатор одного номинала не оправдывает поведение схемы в реальных условиях. Линейная модель задержек не учитывает влияние времени нарастания и спада входного сигнала. Таким образом, способом устранения данных проблем является формирование двумерной таблицы с различной нагрузкой и различными фронтами входного сигнала (нелинейная модель задержек). Затем происходит характеристика временного параметра.

Для характеристики ячеек используется инструмент Liberate от Cadence. Он выполняет следующие действия: определяет функциональную информацию, а затем генерирует тестовые входные воздействия, а затем запускает Spectre. После моделирования экстрагируется информация о временах задержек и создаётся электрическое представление, содержащее информацию о времени, мощности и целостности сигнала в стандартном промышленном формате Synopsys Liberty (.lib).

Обратите внимание, что информацию о площади ячейки необходимо будет добавить вручную в lib-файл.

Скрипт для характеристики ячейки

Для того, чтобы охарактеризовать ячейку, используется скрипт, написанный на языке TCL («ти-си-эль», «тикль», Tool Command Language). В скрипте содержится нижеследующие шаги:

1. определение шаблона для задержек, мощности и временных ограничений;
2. установка PVT-значений (пороговое напряжение, напряжение питания и температура окружающей среды для типичного, лучшего и худшего случаев);
3. задание пути к модели, содержащей определение технологических параметров МОП-транзисторов, таких как пороговое напряжение, толщина слоя оксида и так далее, а также положение выводов транзисторов (исток, сток, затвор и подложка);
4. загрузка модели транзисторов и файла соединений, содержащий ячейки, который необходимо охарактеризовать;

5. определение характеристик;

6. запись данных в lib-файл.

По умолчанию в Librate в качестве ограничения максимального числа разветвлений для нагрузки составляет 20.

Практическое занятие № 6. Поток проектирования

В данной работе будет представлен пошаговый подход к выполнению моделирования RTL, синтезу / моделированию вентиляного уровня и, наконец, проектированию макета с использованием автоматического размещения и маршрута SoC Innovus со стандартной библиотекой ячеек TSMC 45 нм.

Моделирование HDL-описания

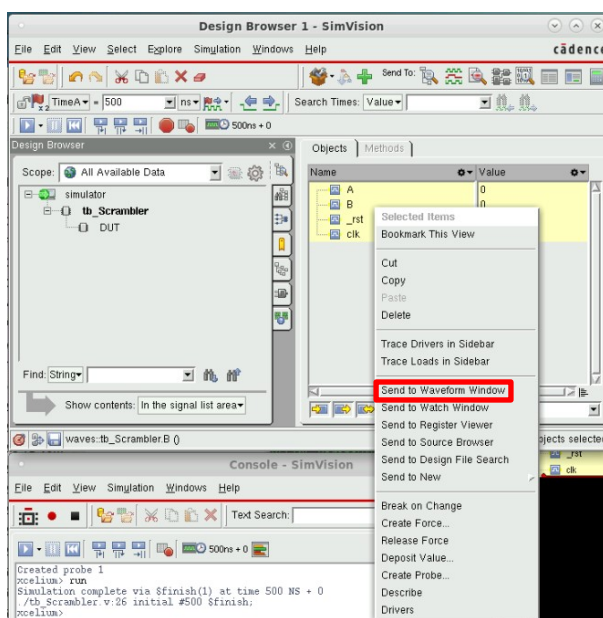
Для запуска моделирования RTL-описания откройте терминал в папке, в которой расположены v-файлы, и запишите команду

```
xmverilog +gui +access+gwc [file1.v] [file2.v]
```

Здесь [file1.v] [file2.v] – наименования v-файлов, которые необходимо использовать в моделировании. Если, например, файл file2.v включён в файл file1.v строкой

```
`include «file2.v»
```

то необходимости указывать этот файл в вышезаписанной команде нет.

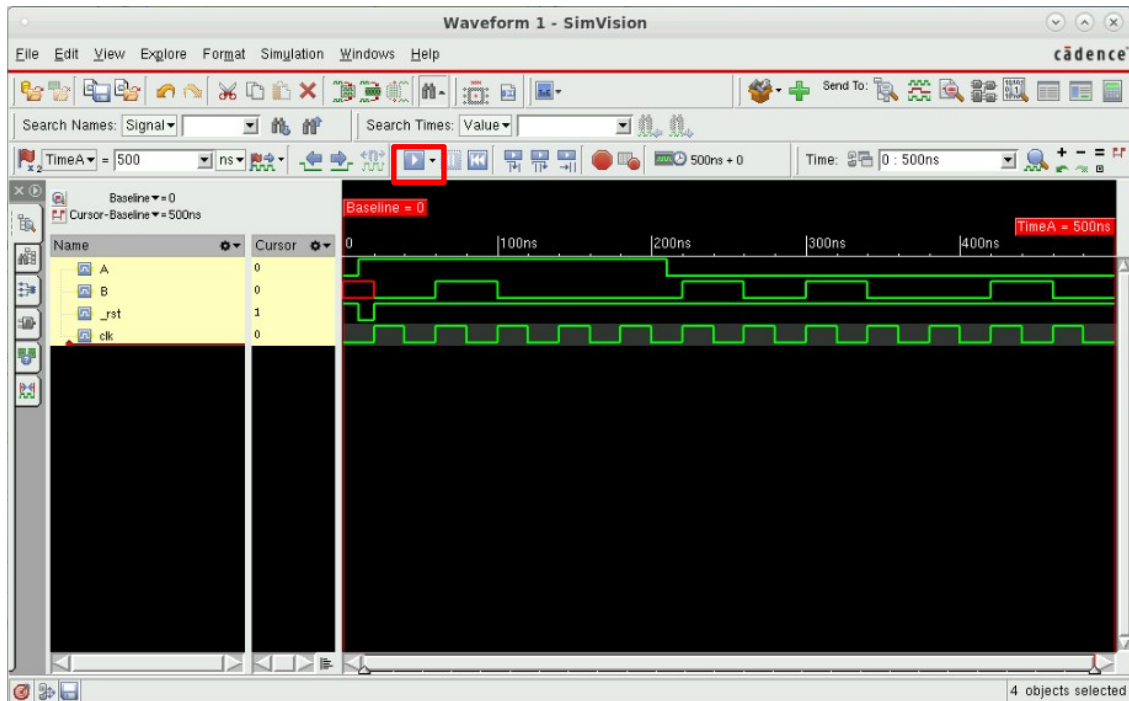


В открывшемся окне модуля SimVision выберите в списке сигналов те, временные диаграммы (waveform) необходимо вывести, нажмите правой кнопкой и в контекстном меню выберите Send to Waveform Editor. Откроется окно вывода временных диаграмм.

В файле тестбенча должно быть указано время, когда необходимо остановить моделирование, например,

```
initial #500 $finish;
```

Далее достаточно запустить процесс моделирования **Simulation > Run** (клавиша F2).



Синтез логической схемы

Следующим этапом проектирования цифровой интегральной схемы является её логический синтез. Задачей логического синтеза является перевод текстового описания работы схемы в представление в виде логических вентилях. Логический синтез осуществляется в базе тех стандартных ячеек, которые присутствуют у целевой библиотеки. В нашем случае используется учебная библиотека от Cadence.

Синтез на уровне вентилях включает в себя реализацию поведения схемы (описываемой моделью Verilog) с использованием стандартных вентилях. При синтезе на уровне вентилях файл Verilog синтезируется в файл списка соединений (netlist), который включает стандартные вентилях и их времена задержек. В соответствии с этим для проведения синтеза необходимо иметь как минимум два типа файлов:

- .v-файл – файл, написанный на языке Verilog HDL, в котором представлено RTL-описание синтезируемого устройства;

- .lib-файл – файл в формате Liberty Timing file в котором содержится информация о стандартных ячейках используемой библиотеки. По сути это файл временной модели, в котором содержится информация о временах задержек ячеек, длительностях фронтов переключения, требования к временам предустановки и удержания. Данный файл предоставляется поставщиком.

В результате синтеза на выходе получится список соединений стандартных ячеек также в формате Verilog, то есть .v-файл. Для различения RTL-описания и описания списка соединений последний некоторые разработчики переименовывают в .vg-файл (равно как для testbench можно использовать .vt-файл).

В любом процессе проектирования моделирование этого списка соединений на уровне вентилях в конечном итоге покажет успешен или нет разрабатываемый проект. Наряду с файлом списка соединений сценарий синтеза также генерирует отчёт о времени задержек (timing report), отчёт о площади и отчёт о потребляемой мощности. Отчёт о временах задержек является важной частью информации. В нём содержится подробная информация о критическом пути, который указывает, соответствует ли схема требованиям к синхронизации / частоте.

Для синтеза используется GENUS Synthesis Solution следует проделать следующие шаги.

1 Создайте в папке с проектом папки /library, /rtl, /syn и /virtuoso

```
mkdir library rtl syn virtuoso
```

2 Сформируйте символичные ссылки на библиотеки с логическими вентилями и контактными площадками в папку /library

```
$ ln -s /local/1/lan/flow/t1u1/reference_libs/GPDK045/gsclib045_all_v4.4/gsclib045/timing/fast_vdd1v0_basicCells.lib ./library/gsclib045_fast.lib
```

```
$ ln -s /local/1/lan/flow/t1u1/reference_libs/GPDK045/gsclib045_all_v4.4/gsclib045/verilog/fast_vdd1v0_basicCells.v ./library/gsclib045_fast.v
```

Для подготовки проекта для размещения и трассировки кристалла нужны дополнительно файлы:

- .lef-файл – файлы формата Library Exchange Format. Файлы, описывающие библиотеку имеют две части: технологическую и описание ячеек. Технологический файл содержит информацию о

доступных слоях металла, сквозных отверстиях (via), правилах проектирования. Файл описания ячеек содержит информацию, относящуюся к геометрии каждой ячейки в абстрактном её представлении.

Перейдите в папку с проектом, а затем при помощи терминала создайте в папке `/library` папку `/lef`, в которой добавьте символные ссылки на lef-файлы (их три: технологический файл `gsclib045_tech.lef`, описание стандартных ячеек `gsclib045_macro.lef` и описание ячеек контактных площадок `giolib045.lef`)

```
$ ln -s /local/2/lan/flow/t1u1/reference_libs/GPDK045/gsclib045_all_v4.4/gsclib045/lef/gsclib045_tech.lef
./library/lef/gsclib045_tech.lef
```

```
$ ln -s /local/2/lan/flow/t1u1/reference_libs/GPDK045/gsclib045_all_v4.4/gsclib045/lef/gsclib045_macro.lef
./library/lef/gsclib045_macro.lef
```

```
$ ln -s /local/3/lan/flow/t1u1/reference_libs/GPDK045/giolib045_v3.2/giolib045/lef/giolib045.lef
./library/lef/giolib045.lef
```

3 Перейдите в папку `/syn` и запустите из неё Genus командой

```
cd /syn
```

```
genus -legacy_ui
```

```
gui_show
```

4 Сверните окно графического интерфейса и проделайте следующие инструкции, чтобы загрузить корректную библиотеку стандартных ячеек и запустить синтез (вместо `<file.v>` укажите файл своего проекта)

```
legacy_genus:/> set OUTPUT_DIR ./run_dir
```

```
legacy_genus:/> set_attribute lib_search_path ../library
```

```
legacy_genus:/> set_attribute library {gsclib045_fast.lib}
```

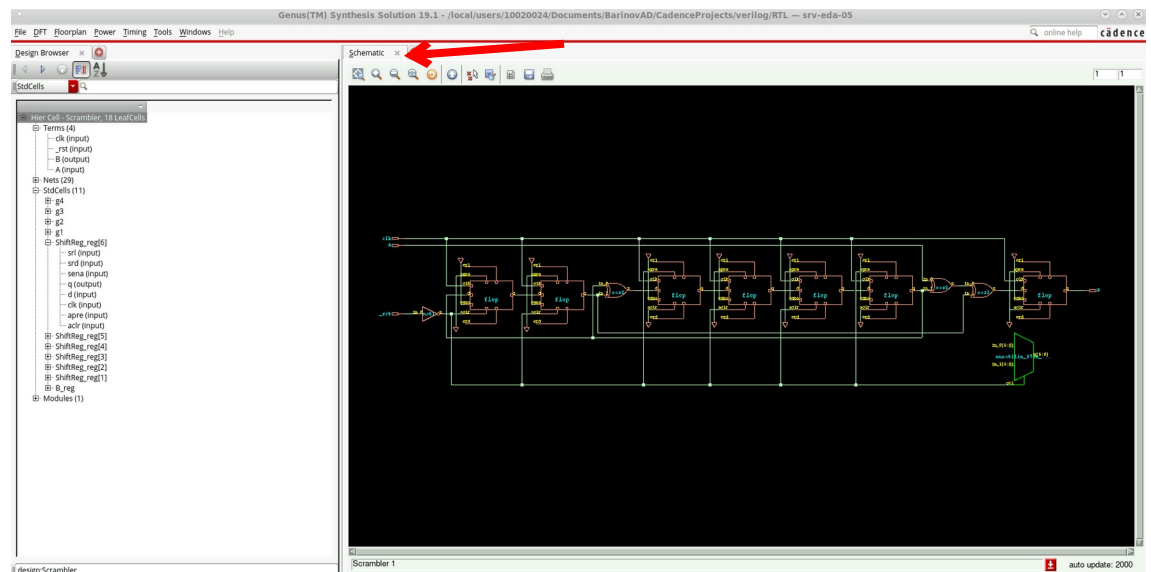
5 Вы получите несколько предупреждений, но ошибок быть не должно. Как только библиотека настроена, вам нужно загрузить и доработать свой файл Verilog. В этом случае он был помещен в папку `/rtl` и может быть обработан с помощью следующих команд (вместо `file` укажите название своего проекта):

```
legacy_genus:/> load -v2001 ../rtl/<file.v>
```

```
legacy_genus:/> set DESIGN <file>
```

```
legacy_genus:/> elaborate
```

6 При доработке Verilog-описания каждая строка текста обрабатывается и заменяется вентилями. Перейдите в окно вашего графического интерфейса, там должен быть пустое представление “Layout”, добавьте вид “Schematic”. Вы должны получить окно с блоками схем, которые представляют входной Verilog-файл.



7 На данный момент компилятор RTL сгенерировал неоптимизированную версию предоставленного Вами Verilog-описания. Заключительным шагом является синтез разработанного дизайна, который оптимизирует уже созданную схему и интегрирует ваш проект со стандартной библиотекой ячеек TSMC. Вариантов оптимизации доступно множество, но в данной работе они приводиться не будут. Проект может быть синтезирован с помощью следующей команды:

```
legacy_genus:/> synthesize -to_mapped -eff high -no_incr
```

Окно графического интерфейса не будет автоматически обновляться. Чтобы обновить графический интерфейс, Вам нужно выбрать свой проект в левом меню (он будет называться Hier Cell), затем щелкнуть правой кнопкой мыши и выбрать **Schematic View > In Main**.

8 После сообщения об успешном синтезе сохраните список соединений и временные ограничения (constraints)

```
legacy_genus:/> write -mapped > ${DESIGN}_synth.v
```

```
legacy_genus:/> write_sdc > ${DESIGN}.sdc
```

9 Выйдите из Genus

```
legacy_genus:/> exit
```


Для синтеза может использоваться скрипт, написанный на языке Tcl (от англ. Tool Command Language — «командный язык инструментов»), который содержит следующие основные указания:

- файл Verilog, содержащий определение модуля верхнего уровня;
- правильный каталог;
- имя модуля верхнего уровня;
- частота, для которой должен быть реализован проект. На основе предоставленной информации о частоте инструмент синтеза пытается оптимизировать конструкцию таким образом, чтобы соответствовать требованиям к этой частоте.

Синтез с использованием скрипта

Некоторые операции при синтезе и разводке кристалла приходится повторять неоднократно. Для того, чтобы ускорить и автоматизировать процесс проектирования кристалла рекомендуется использовать скрипты, написанные на языке Tcl.

Перейдите в папку /syn, в ней создайте файл скрипта для автоматизированного синтеза syn.tcl, содержание файла представлено ниже.

```
pluma syn.tcl
```

В открывшемся редакторе введите текст скрипта. Описание скрипта syn.tcl находится в Приложении А.

Для автоматизации синтеза запустите из папки /syn инструмент Genus в режиме legacy без графического интерфейса, затем запустите скрипт

```
genus -legacy_ui  
legacy_genus:/> source syn.tcl
```

Исправляя ошибки, добейтесь завершения синтеза.

В результате исполнения скрипта в папке появится папка /genus_invs_des, в которой содержатся файлы:

```
genus.v – список соединений на языке Verilog  
genus.globals – файл с глобальными настройками для Innovus  
genus.mode – файл с установками для временного анализа
```

genus.mmode.tcl – tcl-скрипт, содержащий задание параметров временного анализа

Размещение и трассировка

Данный этап проводится инструментом Cadence Innovus.

Описание контактных площадок

1 В папке /genus_invs_des откройте файл genus.v, содержащий описание списка соединений после синтеза

```
gedit genus.v
```

2 В открывшемся файле отредактируйте описание списка соединений, добавив в него описание подключения контактных площадок с ядром. Для этого необходимо добавить:

2.a описание модуля верхнего уровня module top, который будет являться Вашим синтезированным устройством;

2.b внутренние проводники, соединяющие контактные площадки с ядром;

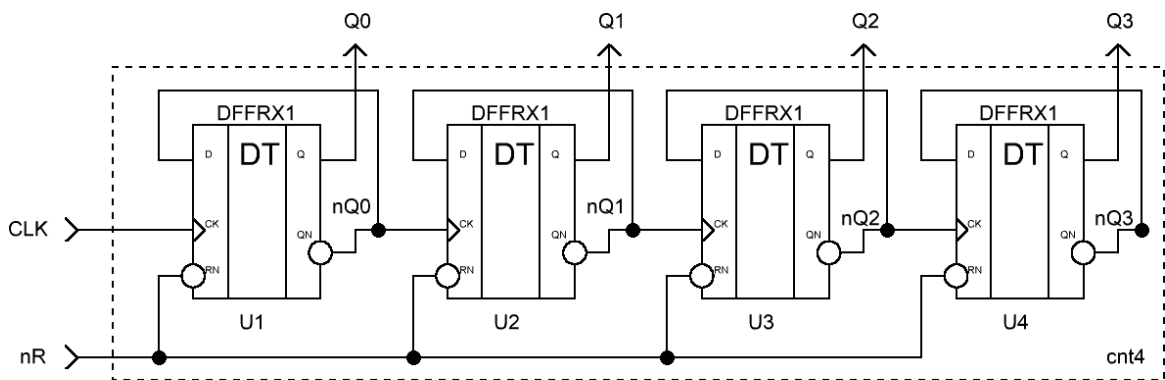
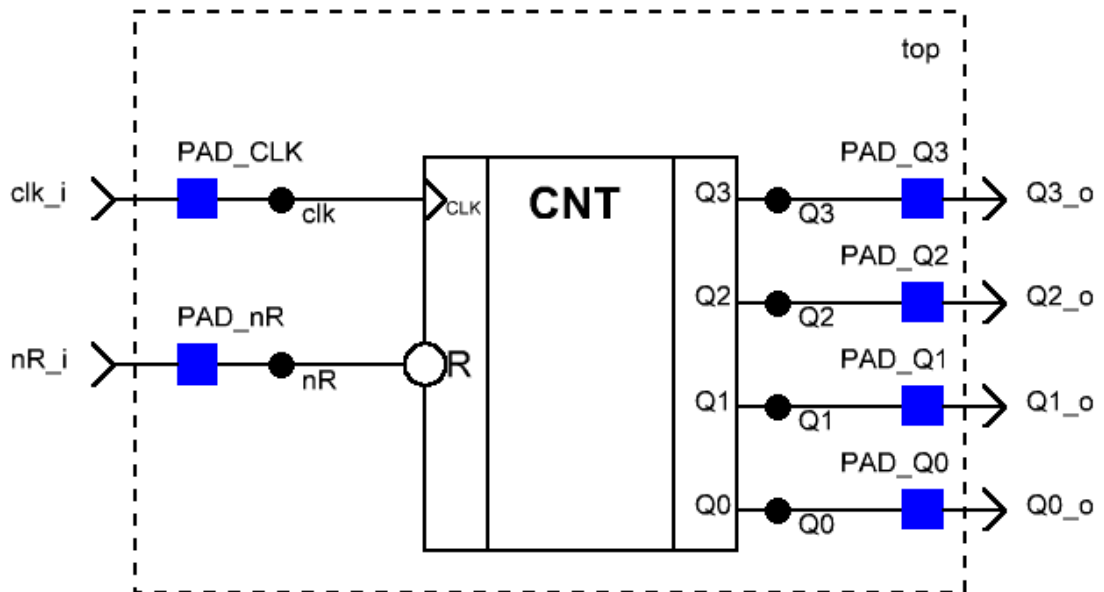
2.c внешние проводники, соединяющие внешнюю часть кристалла с контактными площадками (фиктивно);

2.d проводники, обеспечивающие соединение ядра с питанием и землёй VDDC и VSSC (C – англ. core - ядро);

2.e проводники, соединяющие питание контактных площадок (в рамках используемого PDK, контактные площадки имеют собственное питание) VDDO и VSSO (O – англ. outdoor – наружный, внешний);

2.f экземпляры модулей контактных площадок: входных, выходных, двунаправленных (последние при необходимости), питания ядра и питания контактных площадок и их соединения проводниками;

Для этого используйте пример ниже. В качестве примера используем 4-битный счётчик со сбросом. У него есть 2 контактные площадки для входа (для сигналов clk и nR) и 4 контактные площадки на выход (для сигналов Q3, Q2, Q1 и Q0). Точками указаны провода (wire), которыми соединяются контактные площадки и схема.



Файл Verilog с описанием такого модуля выглядит следующим образом

//описания модулей более низкого уровня иерархии располагаются выше

//описание модуля cnt4

```
module cnt4(CLK, nR, Q3, Q2, Q1, Q0);
```

```
input CLK, nR;
```

```
output Q3, Q2, Q1, Q0;
```

//внутренние межсоединения

```
wire nQ3, nQ2, nQ1, nQ0;
```

//соединения триггеров

```
DFFRX1 U1 (.Q(Q0), .QN(nQ0), .D(nQ0), .RN(nR), .CK(CLK));
```

```
DFFRX1 U2 (.Q(Q1), .QN(nQ1), .D(nQ1), .RN(nR), .CK(nQ0));
```

```
DFFRX1 U3 (.Q(Q2), .QN(nQ2), .D(nQ2), .RN(nR), .CK(nQ1));
```

```

DFFRX1 U4 (.Q(Q3), .QN(nQ3), .D(nQ3), .RN(nR), .CK(nQ2));
endmodule

//описываем входы и выходы top-модуля
module top(clk_i, nR_i, Q3_o, Q2_o, Q1_o, Q0_o);
input clk_i, nR_i;
output Q3_o, Q2_o, Q1_o, Q0_o;

//задаём проводники, которые соединят
//счётчик с контактными площадками
wire clk, nR, Q3, Q2, Q1, Q0;

//задаём проводники для подключения к питанию ядра (VDDC, VSSC)
//и контактных площадок (VDDO, VSSO)
wire VDDC, VDDO;
wire VSSC, VSSO;

//подключаем счётчик к межсоединениям
cnt4 U1 (.clk(clk),
        .R(nR),
        .Q3(Q3),
        .Q2(Q2),
        .Q1(Q1),
        .Q0(Q0));

//Входные контактные площадки
PADDI                                     PAD_CLK
(.PAD(clk_i), .Y(clk), .VDDIOR(VDDO), .VSSIOR(VSSO));
PADDI PAD_nR (.PAD(nR_i), .Y(nR), .VDDIOR(VDDO), .VSSIOR(VSSO));

//Выходные контактные площадки
PADD0 PAD_Q3 (.A(Q3), .PAD(Q3_o), .VDDIOR(VDDO), .VSSIOR(VSSO));
PADD0 PAD_Q2 (.A(Q2), .PAD(Q2_o), .VDDIOR(VDDO), .VSSIOR(VSSO));
PADD0 PAD_Q1 (.A(Q1), .PAD(Q1_o), .VDDIOR(VDDO), .VSSIOR(VSSO));
PADD0 PAD_Q0 (.A(Q0), .PAD(Q0_o), .VDDIOR(VDDO), .VSSIOR(VSSO));

//Контактные площадки питания ядра
//например, для больших схем по 2 для VDD и VSS
PADVDD PAD_VDD1 (.VDD(VDDC), .VDDIOR(VDDO), .VSSIOR(VSSO));
PADVDD PAD_VDD2 (.VDD(VDDC), .VDDIOR(VDDO), .VSSIOR(VSSO));
PADVDD PAD_VSS1 (.VSS(VSSC), .VDDIOR(VDDO), .VSSIOR(VSSO));
PADVDD PAD_VSS2 (.VSS(VSSC), .VDDIOR(VDDO), .VSSIOR(VSSO));
//Контактные площадки питания контактных площадок входа/выхода
PADVDDIOR PAD_VDDIOR (.VDDIOR(VDDO), .VSSIOR(VSSO));
PADVSSIOR PAD_VSSIOR (.VDDIOR(VDDO), .VSSIOR(VSSO));

```

Создание проекта

1 Перейдите в папку, в которой размещается синтезированный проект, и запустите в ней модуль Innovus командой в терминале

```
innovus
```

2 Запустите в открывшемся окне импорт проекта **File > Import Design**.

1.a В открывшемся окне в блоке *Netlist* выберите Verilog и выберите файл со списком соединений `genus_invs_des/genus.v`.

1.b В поле Top Cell выберите By User и укажите наименование модуля верхнего уровня top.

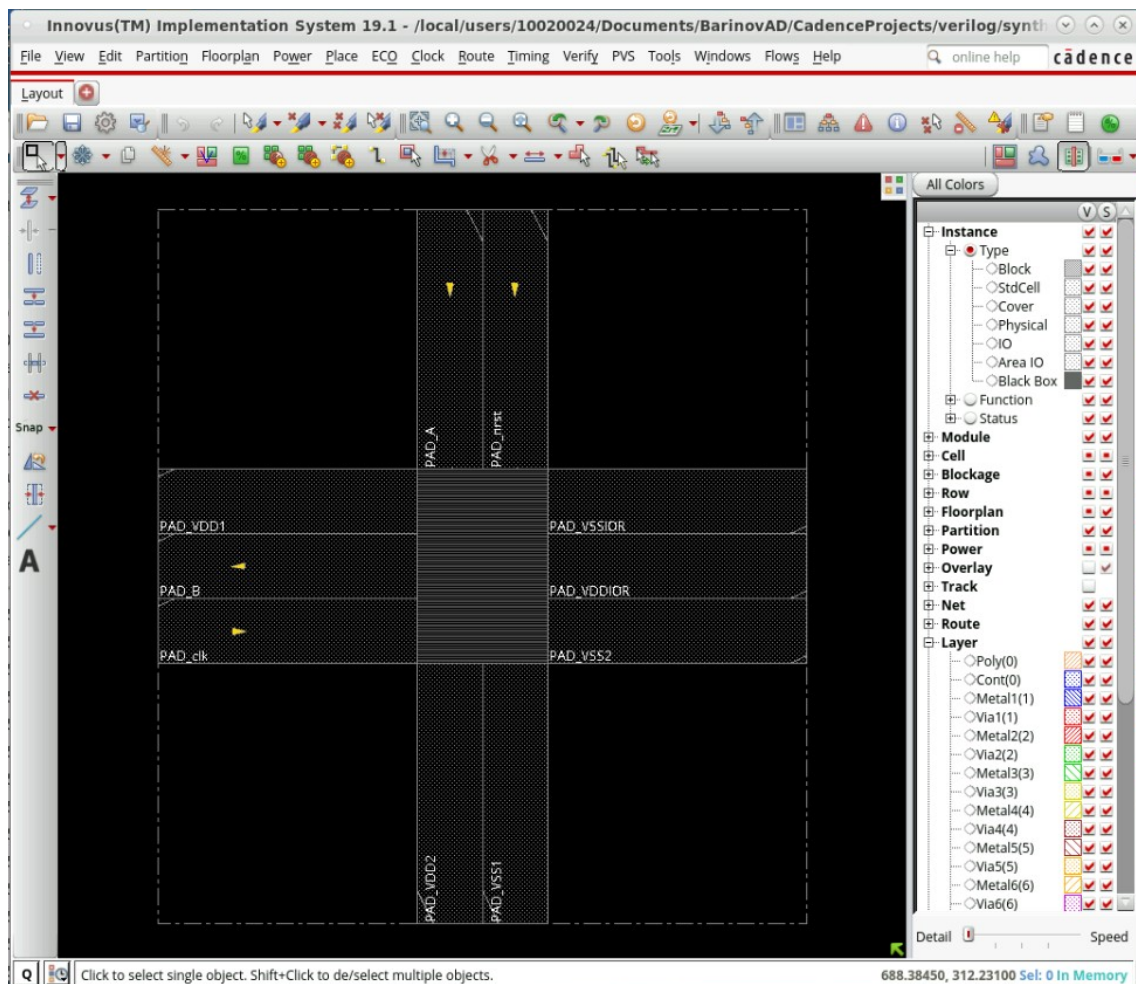
1.c В блоке *Technology/Physical Libraries* укажите LEF Files и выберите LEF-файлы из папки с библиотекой `library/lef/gsclib045_tech.lef`, `library/lef/gsclib045_macro.lef` и `library/lef/giolib045.lef`. **Порядок перечисления важен!**

1.d В блоке *Power* укажите наименования шин питания и земли для ядра и контактных площадок VDDC, VDDO и VSSC с VSSO соответственно (см. пункт про формирование описания контактных площадок).

1.e В блоке *Analysis Configuration* в поле MMMC View Defenition File выберите файл `genus_invs_des/genus.mmode.tcl`, в этом файле указаны временные ограничения для проведения оптимального процесса размещения и трассировки.

1.f Нажмите ОК.

1.g Если всё хорошо, то появится изображение кристалла с контактными площадками и пустым ядром (стрелки у контактных площадок входа/выхода показывают входная или выходная это площадка).



3 Сохраните проект, чтобы можно было его загружать без создания заново **File > Save Project**. В открывшемся окне выделите тип данных *Innovus*.

4 Чтобы загрузить проект выберите **File > Restore Design**. В открывшемся окне также выберите тип данных *Innovus*.

Создание дополнительных контактных площадок

На предыдущем рисунке обратите внимание, что контактные площадки расположены хаотично, хоть и равномерно вокруг ядра микросхемы. Кроме того, отсутствуют специальные ячейки, которые обеспечивают соединение контактных площадок по углам – угловые ячейки входа / выхода. Следовательно, на текущем шаге необходимо завершить формирование контактных площадок.

После сохранения проекта в Innovus в папке с проектом появился файл *top.save.io*. Если не появился, выберите **File > Save... > IO Files**. В этом файле находится информация о контактных площадках, их типе, расположении относительно ядра (сверху, справа, снизу, слева), порядок их следования, подключаемые проводники и прочее.

В данном пункте мы отредактируем этот файл с указанием информации о контактных площадках и добавим новые.

Откройте файл top.save.io в текстовом редакторе и отредактируйте его, добавив угловые контактные площадки, по примеру описания ниже

```
#определяем глобальные параметры
#направление расположения площадок - по часовой стрелке
(globals
  version = 3
  io_order = clockwise
)
#описываем площадки (указываем имя, тип ячейки
#и ориентацию для угловых ячеек)
#слева сверху, справа сверху, справа снизу, слева снизу - угловые
#контактные площадки
#слева, сверху, справа и снизу - контактные площадки
#входные / выходные, питания ядра и питания контактных площадок
#имена площадок должны соответствовать описанным в netlist

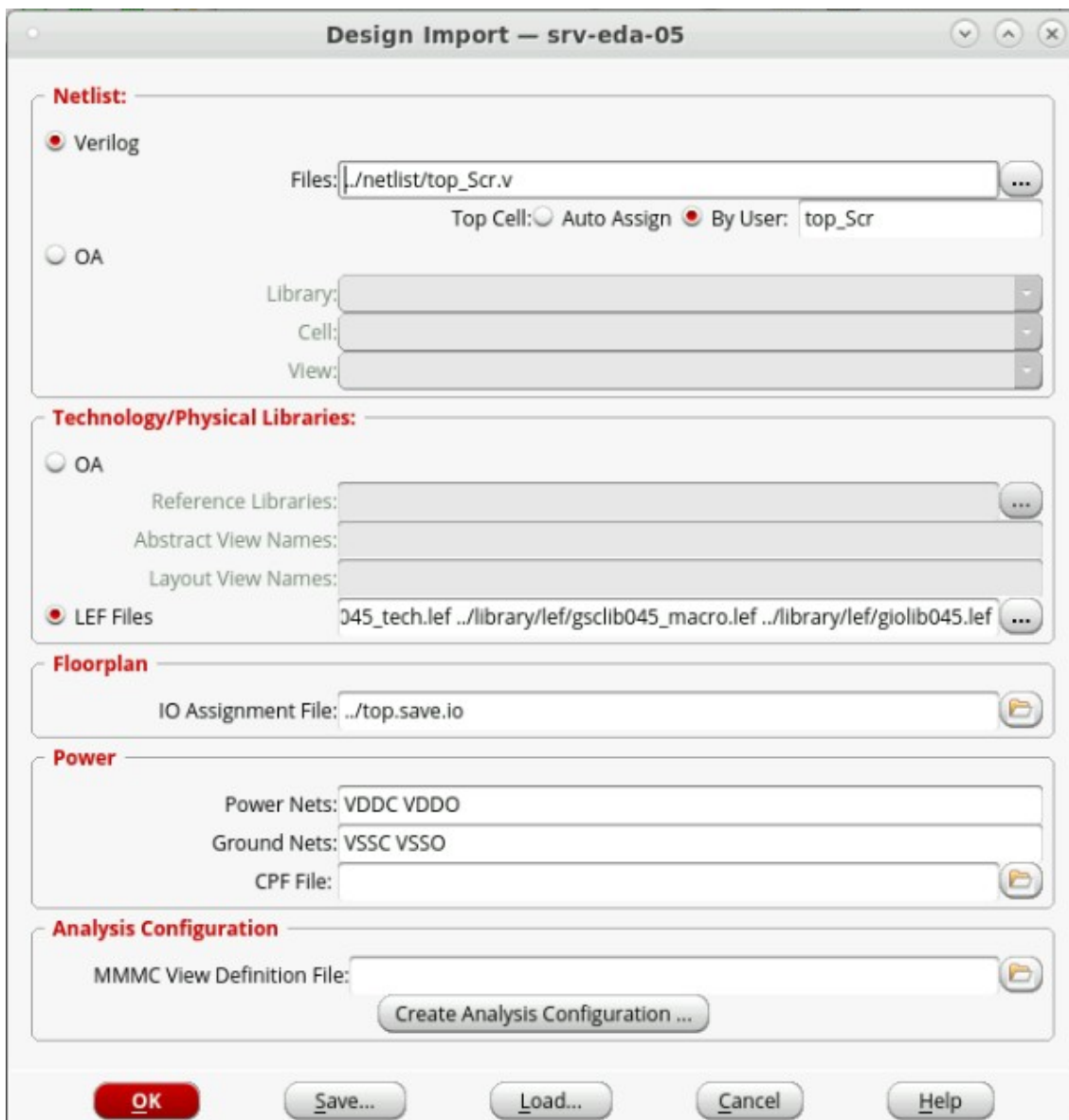
(iopad
#угловая площадка сверху слева
  (topleft
    (inst name="PAD_Corner1" cell=padIORINGCORNER orientation =
R270)
  )
#площадки сверху
  (top
    (inst name="PAD_clk" cell=PADDI)
    (inst name="PAD_VDD1" cell=PADVDD)
  )
#угловая площадка сверху справа
  (topright
    (inst name="PAD_Corner2" cell=padIORINGCORNER orientation =
R180)
  )
#площадки справа
  (right
    (inst name="PAD_Q0" cell=PADD0)
    (inst name="PAD_Q1" cell=PADD0)
    (inst name="PAD_Q2" cell=PADD0)
    (inst name="PAD_Q3" cell=PADD0)
    (inst name="PAD_VSS2" cell=PADVSS)
  )
#угловая площадка снизу справа
  (bottomright
    (inst name="PAD_Corner3" cell=padIORINGCORNER orientation =
R90)
  )
#площадки снизу
  (bottom
    (inst name="PAD_nrst" cell=PADDI)
```

```

        (inst name="PAD_VDD2"      cell=PADVDD)
    )
#угловая площадка снизу слева
    (bottomleft
        (inst name="PAD_Corner4"  cell=padIORINGCORNER orientation =
R0)
    )
#площадки слева
    (left
        (inst name="PAD_VSS1"      cell=PADVSS)
        (inst name="PAD_en"        cell=PADDI)
    )
)

```

Сохраните файл. В следующий раз при загрузке проекта данный файл можно подгружать в окне Import Design в поле IO Assignment (а чтобы не подгружать каждый раз, используйте **Save...** и сохраните файл глобальных настроек в формате *.globals (перезапишите файл genus.globals), который можно в последствие загружать через **Load...**). При необходимости изменить расположение площадок можно отредактировать данный файл.



Планирование кристалла

Для настройки планирования кристалла (размеры ядра или чипа) откройте меню **Floorplan > Specify Floorplan**.

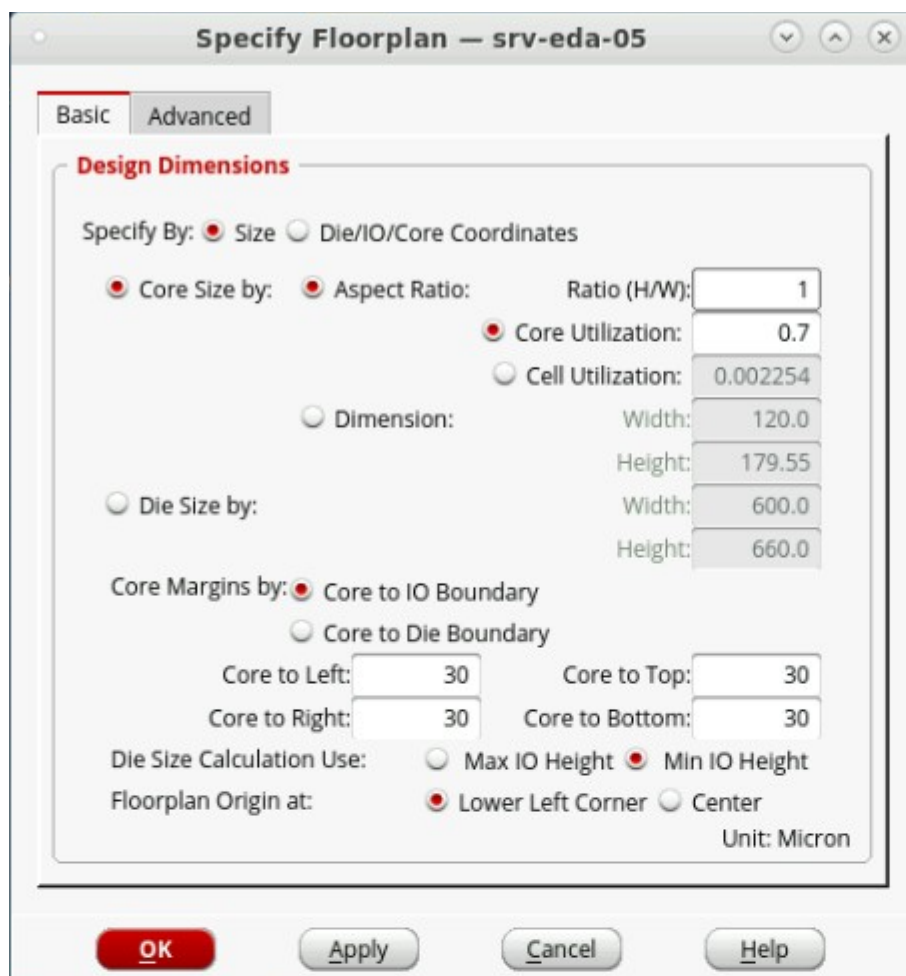
1 В открывшемся окне на вкладке Basic выберите размер ядра Core Size by Aspect Ratio и в поле Ratio (H/W) укажите 1 (соотношение высоты к ширине как у квадрата).

2 Выберите долю использования всей площади ядра Core Utilization равную 0,7 (70 %). Это значение определяет часть площади ядра, используемую для размещения стандартных ячеек. Большее значение означает, что Ваш проект будет довольно компактным, но трассировка может быть сложной, если не невозможной. Меньшие значения значительно увеличат площадь проектирования.

3 Укажите расстояния между ядром и контактными площадками Core Margins by Core to IO Boundary и равными 30 (единица измерения –

микрометры). Чем больше контактных площадок, тем большее расстояние следует указывать. Подбирается опытным путём.

4 Нажмите ОК.



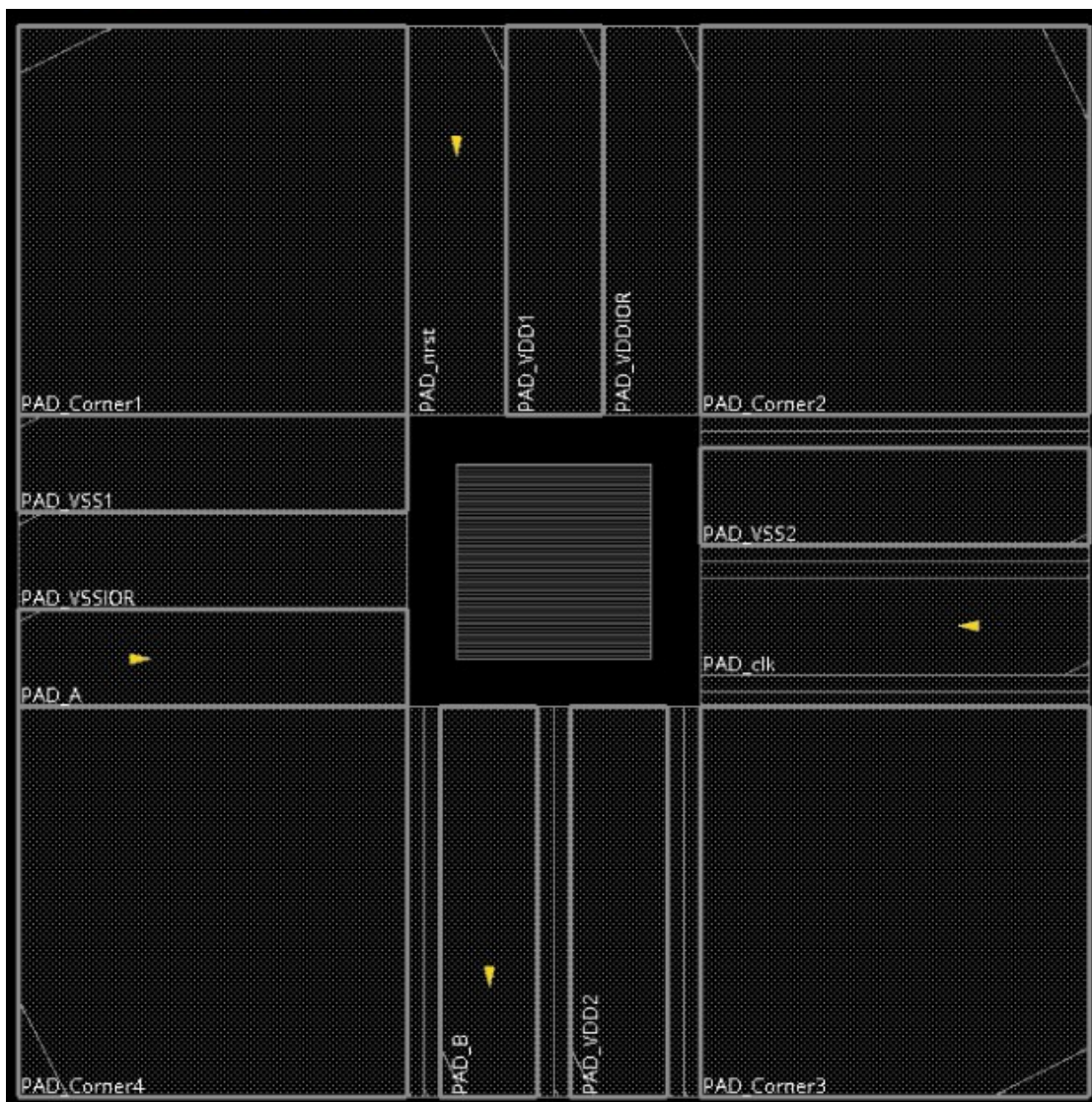
5 После этого получим изображение контактных площадок и ядра.

6 На рисунке видно, что между частью контактных площадок присутствуют пустые места. Их заполняют специальными ячейками – филлерами (fillers). Филлеры в библиотеке giolib045 существуют нескольких шириин - 60, 10, 5, 3, 1, 0,1, 0,01 и 0,005 мкм. Для их создания нажмите **Place > Physical Cell > Add I/O Filler...**

7 В открывшемся окне в качестве имени ячейки введите Cell Name = padIORINGFEED60 (или padIORINGFEED10, или padIORINGFEED5, или padIORINGFEED3, или padIORINGFEED1, или padIORINGFEED01, или padIORINGFEED001, или padIORINGFEED0005). В списке Side выберите ту сторону, где необходимо создать филлеры. Так, для примера, на картинке выше необходимо создать филлеры снизу и справа (bottom и right). Поочерёдно выбирая имена ячеек с шириной от 60 до 1 мкм и нажимая каждый раз Apply убедитесь, что все пустые места заполнены ячейками-филлерами. Для нашего примера понадобились филлеры только шириной 10 мкм.



8 Нажмите ОК.



Создание колец питания и земли

Чтобы подключить Ваше напряжение питания к стандартным ячейкам, питание и земля должны быть доступны со всех сторон матрицы ячеек. Это делается путём добавления колец питания и указания глобальных подключений. Откройте меню **Power > Connect Global Nets**.

1 В области Power Ground Connection в блоке *Connect* выберите Pin, а в блоке *Scope* выберите Apply All.

2 Для каждого сигнала VDDC, VDDO, VSSC и VSSO сделайте следующее:

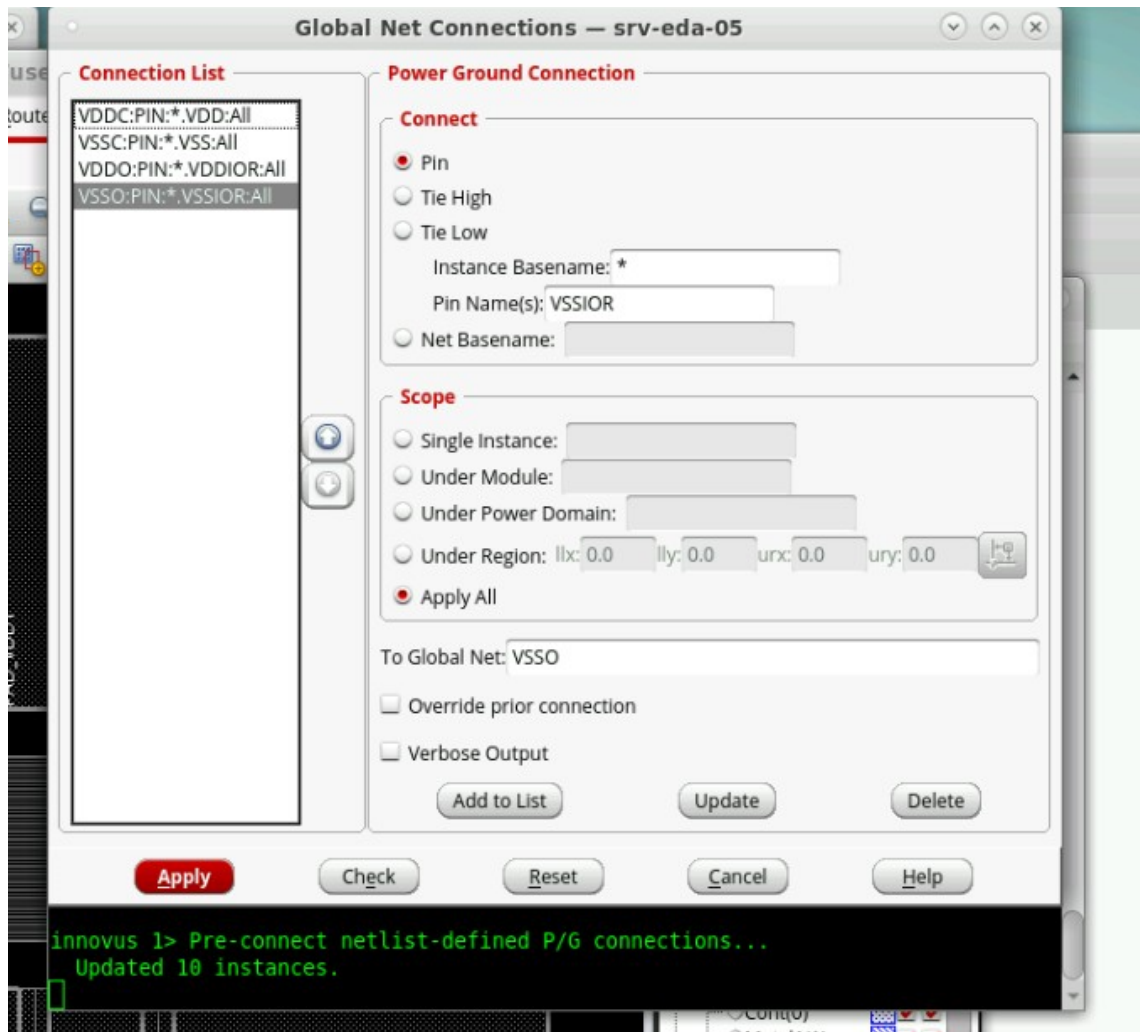
2.a Введите имя пина в поле Pin Name(s) в области *Connect*

2.b Введите имя сигнала в поле To Global Net.

2.c Нажмите *Add to List*.

2.d Так необходимо п. 2, а-с сделать для подсоединения сигналов VDDC к пину VDD, VSSC – к VSS, VDDO – к VDDIOR, VSSO – к VSSIOR.

- 3 Нажмите *Apply* (в терминале отобразится сколько ячеек обновлено), а затем *Check* для проверки, все ли сигналы подсоединены (в терминале после нажатия этой кнопки не будет сообщений, если всё хорошо).
- 4 Нажмите *Cancel* для выхода.

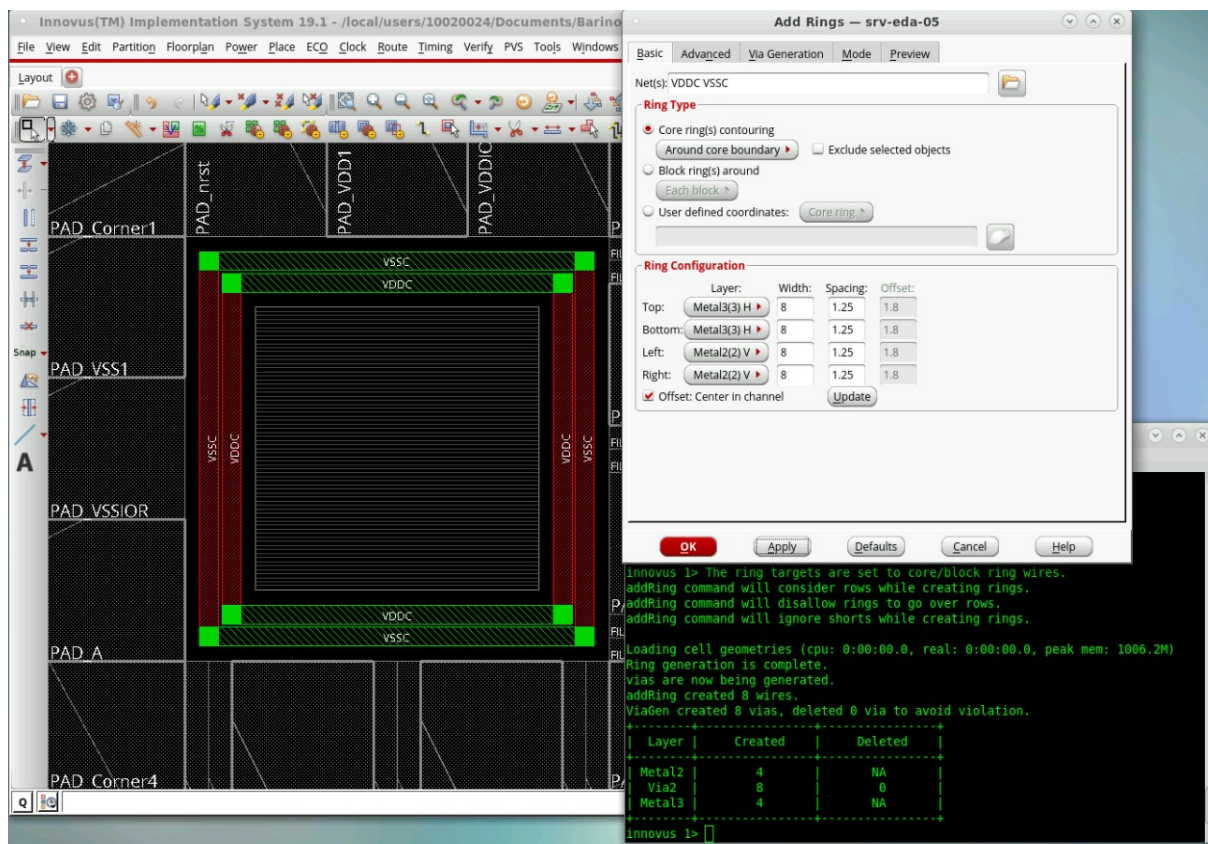


- 5 Затем выберите **Power > Power Planning > Add Ring**.
- 6 В открывшемся окне в поле Net(s) добавьте VDDC и VSSC, кликнув на иконку папки справа.
- 7 Вы можете планировать питание в различных режимах, чтобы облегчить различные стили дизайна. Поскольку предпочтительнее выполнять маршрутизацию внутри колец питания, лучше расположить их вокруг границы ввода-вывода, поэтому выберите этот вариант. В блоке **Ring Type** выберите *Around I/O boundary*.
- 8 В блоке **Ring Configuration** укажите слой Metal3 для Top и Bottom и слой Metal2 для Right и Left. Горизонтальная линия металлизации таким образом будет выполнена в слое Metal3, а вертикальная – в слое Metal2.
 - 8.а Ширину выставьте равную Width = 8 (Расчёт ширины колец представлен ниже).

8.b Расстояние между одинаковыми слоями Spacing = 1.25.

8.c Offset укажите Center in channel.

9 Нажмите ОК. Появятся кольца вокруг ядра, а в терминале отобразится отчёт о количестве проводников каждого слоя и сквозных отверстиях между ними.



10 Теперь необходимо подсоединить кольца питания и земли к стандартным ячейкам. Для этого выберите **Route > Special Route**, в поле Net(s) введите VDDC и VSSC и нажмите ОК. Будут созданы рельсы (rails) питания и земли к области размещения стандартных ячеек.

11 Для обеспечения более равномерного подключения стандартных ячеек к питанию в больших схемах создаются специальные полосы проводников питания (stripes). Для их создания выберите **Power > Power Planning > Add Stripes**.

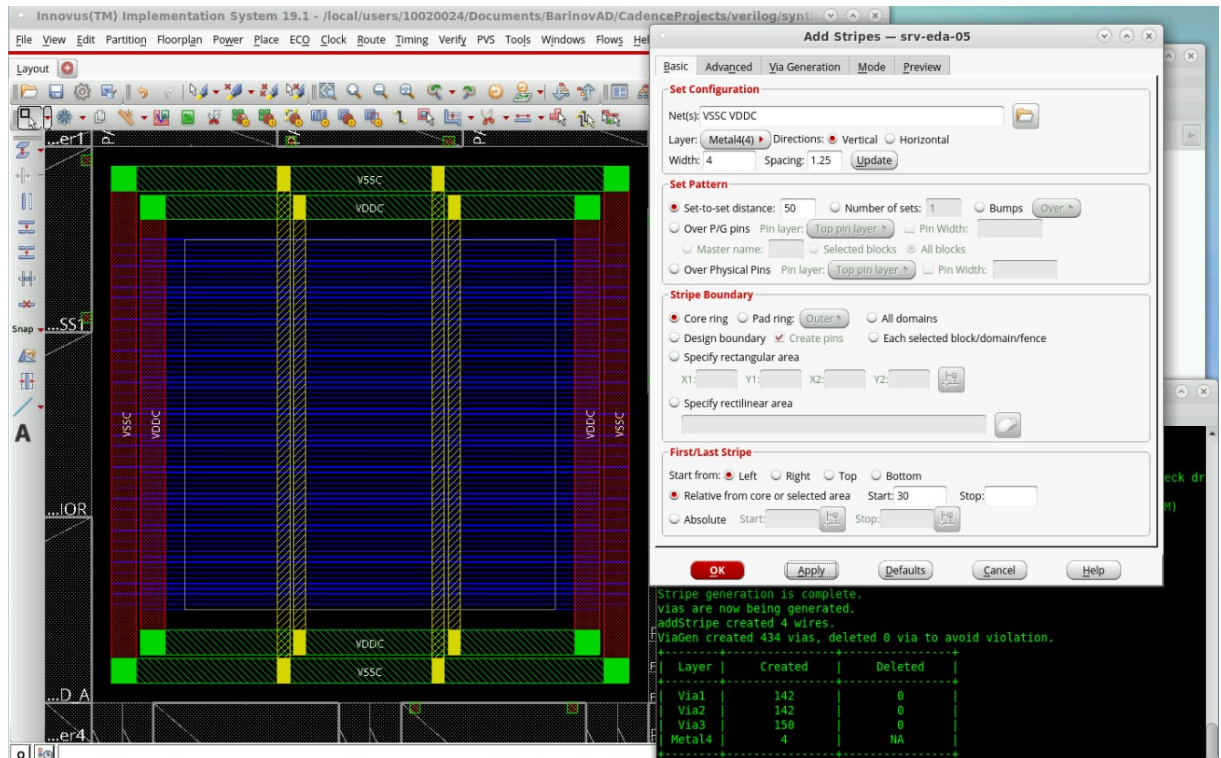
1.a На вкладке **Basic** в поле Net(s) укажите VDDC и VSSC.

1.b В области Set Configuration выберите Layer = Metal4, а направление укажите вертикальное Direction = Vertical.

1.c Ширину Width и минимальное расстояние Spacing между слоями укажите равную половине ширины шин в кольце и 1.25 соответственно.

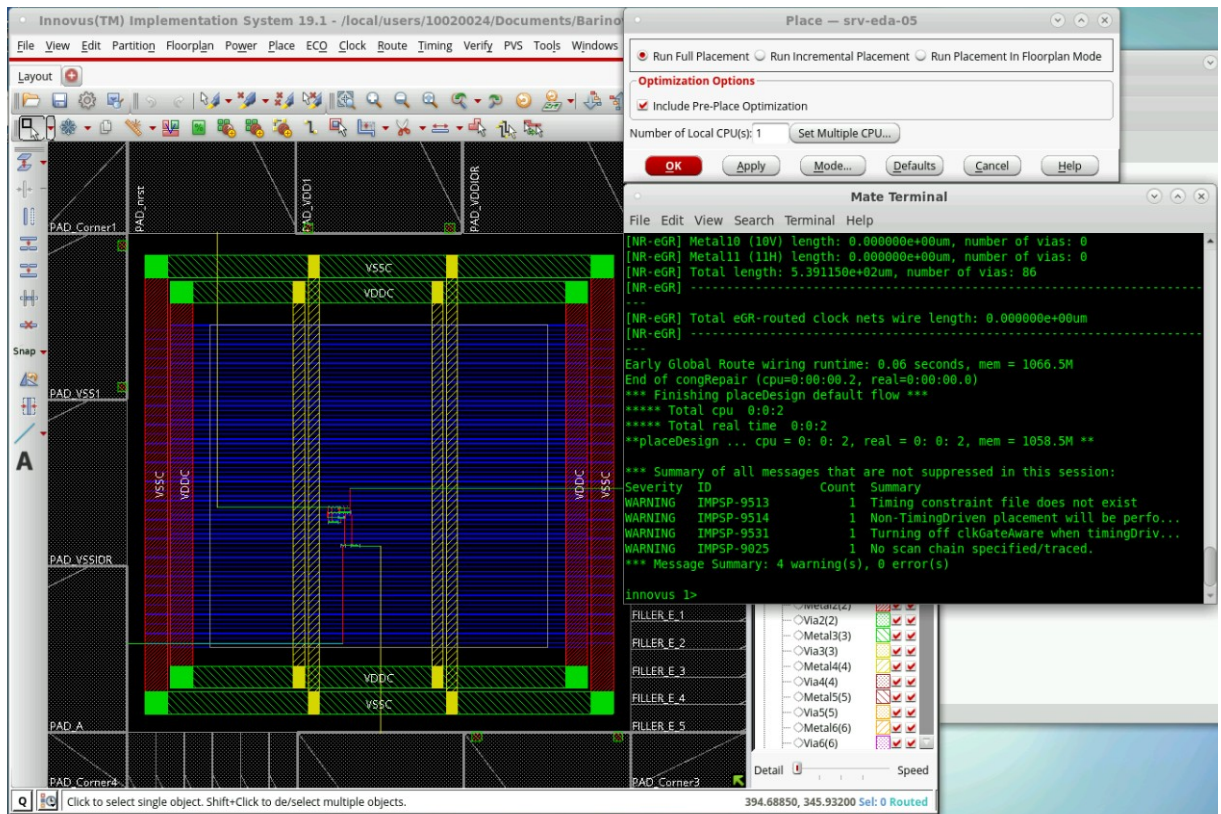
1.d В блоке **Set Pattern** укажите расстояние между полосами Set-to-set distance = 50 (при необходимости можно варьировать).

- 1.e В блоке **First/Last Stripe** выберите **Relative from core or selected area** и укажите **Start = 30** (при необходимости можно варьировать).
- 1.f На вкладке **Advanced** установите **Snap wire center to routing grid = Grid**.
- 1.g Нажмите **OK**. На плане кристалла появятся вертикальные полосы питания и земли внутри ядра.



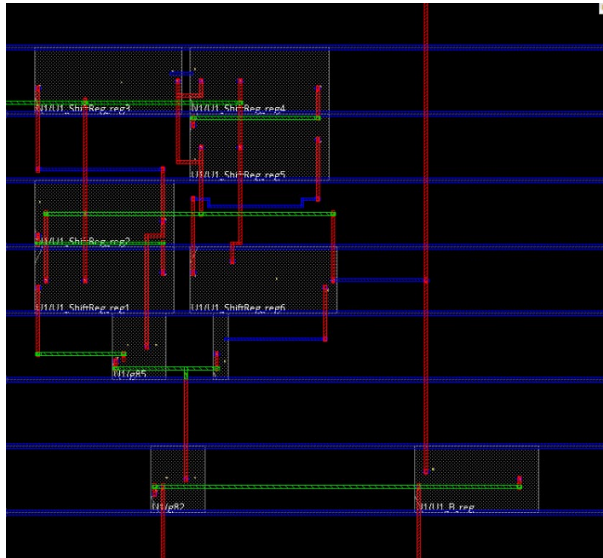
Размещение и трассировка стандартных ячеек

1 Для размещения стандартных ячеек выберите **Place > Place Standart Cell**. В открывшемся окне выберите **Run Full Placement** и нажмите **OK**. Внутри ядра появятся Ваши ячейки с предварительным межсоединением и соединением с контактными площадками.



2 Если размещение завершается неудачей, это, скорее всего, связано с тем, что Innovus не может разместить ячейки в заданной области, не вызывая перекрытия вентилях. Чтобы решить эту проблему, следует начать сначала, повторно инициализировав планирование кристалла с ослабленными ограничениями. Наилучшим ограничением для изменения является коэффициент использования строк. Уменьшив этот коэффициент, Вы сможете увеличить вероятность размещения без перекрытий.

3 Для оптимизации трассировки межсоединений внутри ядра выберите **Route > Nano Route > Route**. В открывшемся окне нажмите **OK**.



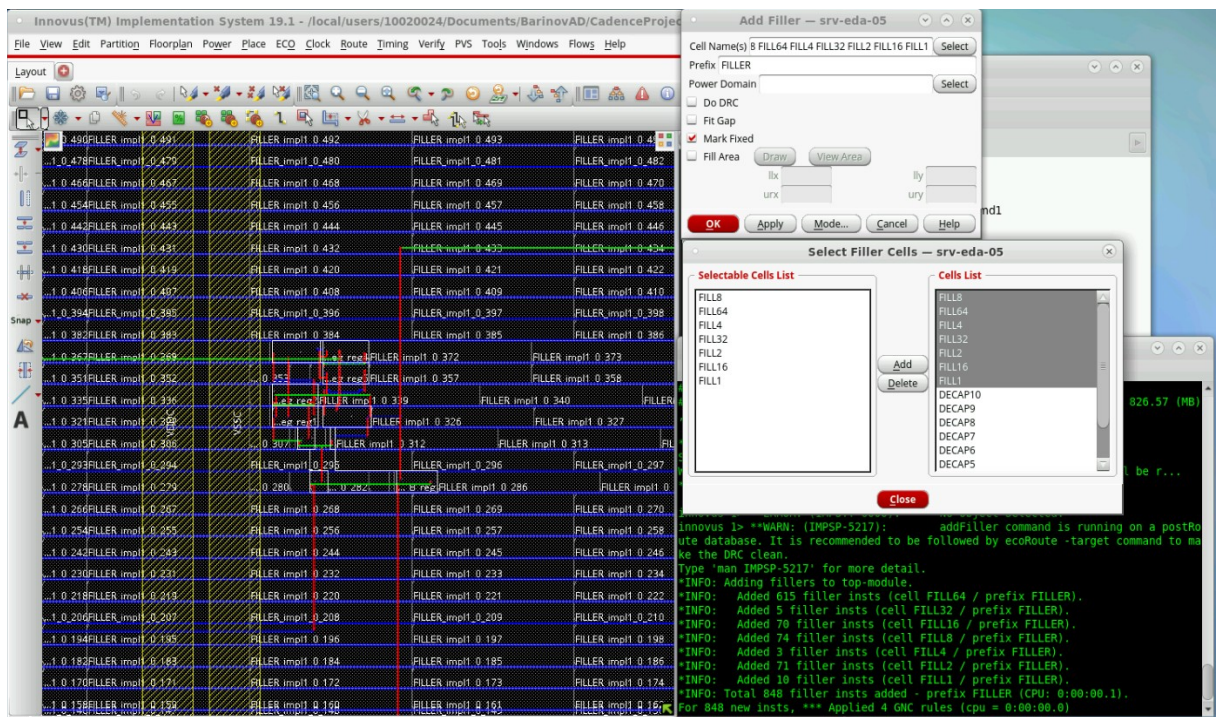
Оптимизация размещения

Если вы импортируете полный макет из Innovus в Virtuoso на этом этапе, у Вас могут возникнуть ошибки DRC из-за того, что слои n -карманов расположены недостаточно далеко. Эти ошибки можно исправить вручную в Virtuoso или добавить предварительно разработанные ячейки-филлеры с помощью Innovus. Эти ячейки обеспечивают непрерывность для слоев n -карманов и линий питания между размещенными ячейками.

Добавление филлеров

Для добавления филлеров проделайте следующие шаги.

- 1 Выберите **Place > Physical Cell > Add Filler**.
- 2 В открывшемся окне в поле Cell Name(s) введите имена ячеек-филлеров, которые будут использоваться для заполнения, для этого нажмите *Select*, выберите все филлеры с именем FILL и нажмите *Add*, нажмите *Close*, отметьте Mark Fixed.
- 3 Нажмите *OK*.
- 4 Убедитесь, что филлеры добавлены в топологию.



Создание топологии завершено!

Следующим этапом является получение DEF-файла (Design Exchange Format), с последующей работой с ним в Virtuoso, где будет происходить верификация топологии, экстракция паразитных параметров и моделирование работы схемы с ними. После данной верификации уже получают набор фотошаблонов в формате GDSII – gds-файл.

Верификация топологии

Перед экспортом макета необходимо убедиться в отсутствии каких-либо геометрических ошибок, возникших во время размещения, или ошибок подключения из-за маршрутизации. Это можно сделать, выбрав пункт **Verify Geometry** и **Verify Connectivity** в меню **Verify**. Отчёты будут сгенерированы и помещены в каталог, из которого Вы запустили Innovus. Вы также можете их увидеть в терминале.

Экспорт DEF-файла из Innovus

Для работы с верификацией топологии, которая производится инструментом Virtuoso необходимо создать специальный файл:

- .def-файл – файл формата Design Exchange Format, который используется для представления физического представления микросхемы. Файл, тесно связанный с LEF-файлом. Таким

образом, оба файла необходимы для правильного отображения физического представления проекта.

В окне Innovus выберите **File > Save > DEF**. Сохраните Вашу топологию в .def-файле.

Пример описания tcl-скрипта для полуавтоматизированной (некоторые размеры необходимо править вручную) разводки кристалла и получения DEF-файла представлен в Приложении Б.

Импорт DEF-файла в Virtuoso и получение layout-представления кристалла

1 Создайте папку /virtuoso в папке с проектом. В ней создайте файл cds.lib, откройте его текстовым редактором и добавьте библиотеки с технологией gpdk045, со стандартными ячейками gsclib045 и контактными площадками giolib045, записав следующее

```
INCLUDE [путь к папке]/gpdk045_v_6_0/cds.lib
```

```
INCLUDE [путь к папке]/gsclib045_all_v4.4/cds.lib
```

```
INCLUDE [путь к папке]/giolib045_v3.2/cds.lib
```

2 Откройте Virtuoso и создайте новую библиотеку для Вашего проекта **File > New > Library**. При создании библиотеки выберите Attach to an existing technology library и укажите gsclib045_tech.

3 Импортируйте библиотеку **File > Import > DEF**. Укажите импортируемый def-файл, библиотеку и ячейку, в которую будет производиться импорт, создаваемое представление как layout. В списке технологических библиотек укажите только gsclib045_tech. В качестве представлений компонентов также укажите layout. Нажмите ОК. В результате импорта Вы получите кристалл в layout-представлении.

DEF In

DEFIn File Name ...

Target Library Name ▾

Target Library Path ...

Ref. Technology Libraries

Create a module hierarchy from hierarchical names Share Library

New Library

Technology From Library ▾

Target Cell Name

Target View Name

Component View List

Master Library List

Overwrite Design Create CustomVias only

Log File Name

Use Template File Use GUI Fields

Template File Name

Save Template File Name ...

Comment Char

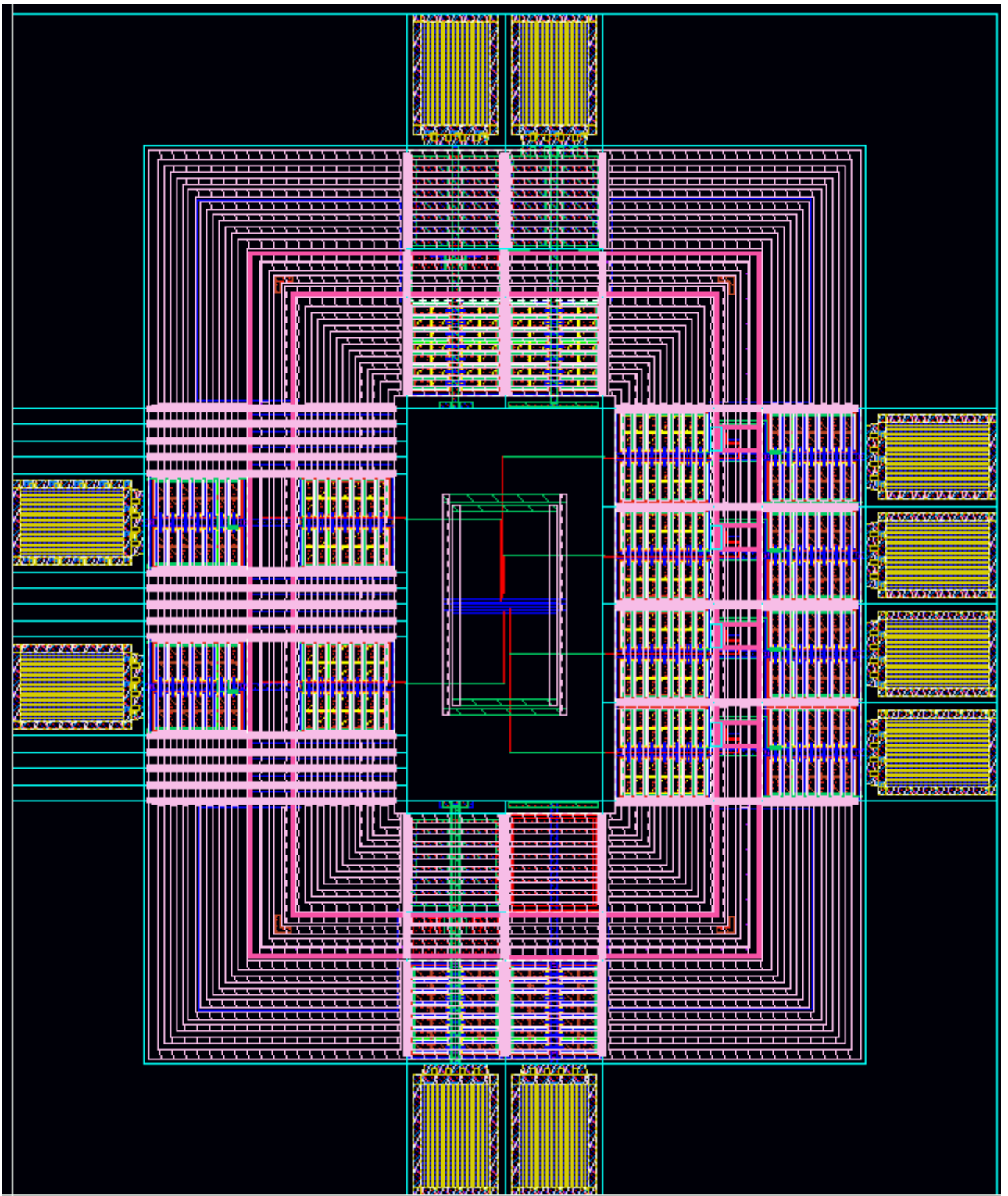
Pin Purpose

Do not create any routing data

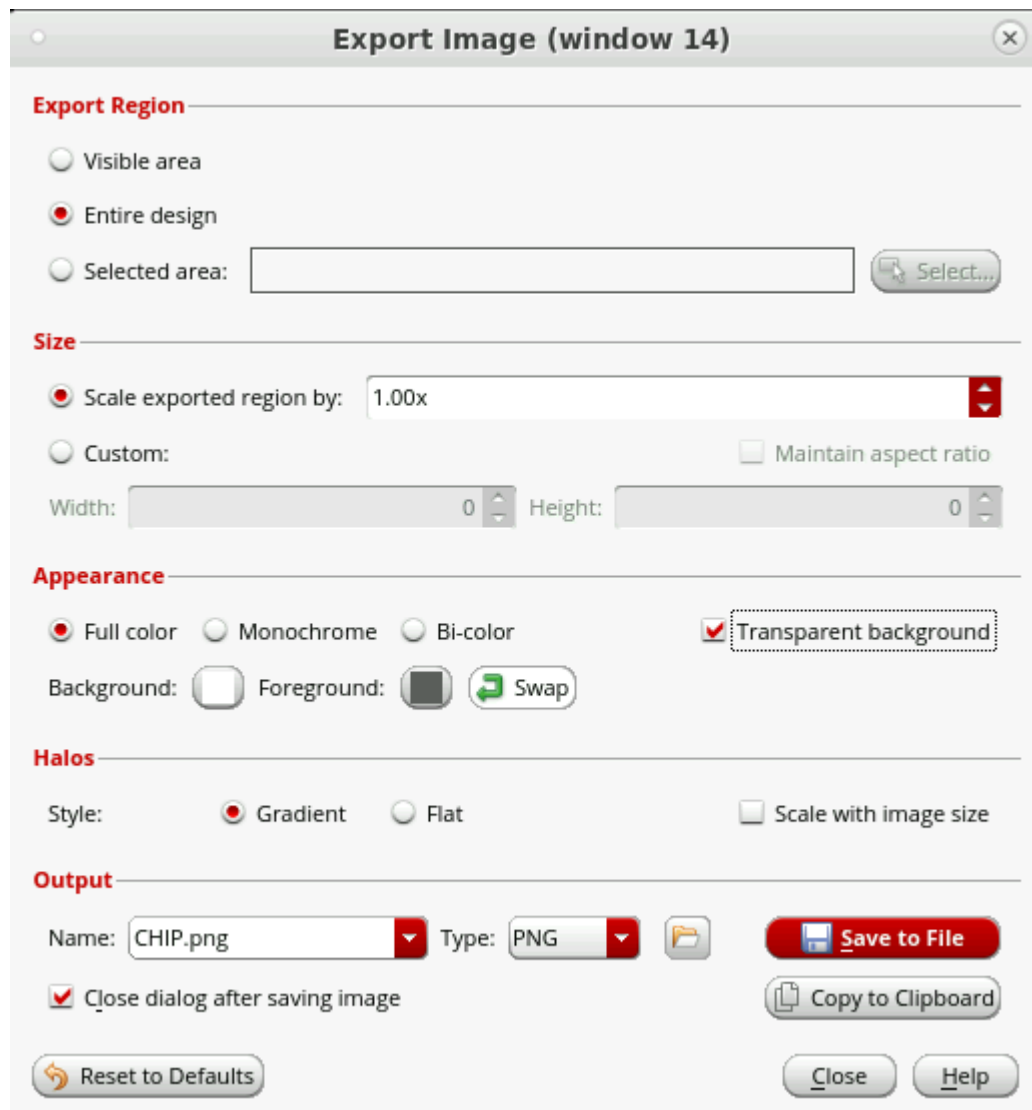
Layer Map File Name

Compress Compress Level

Ignore DRCFILL Shape Tag & Translate on Drawing Purp



4 Экспортировать изображение топологии можно выбрав меню **File > Export Image...** В открывшемся окне настраивается экспортируемая область, формат файла и пр. Для представления изображения для отчётов рекомендуется использовать белый либо прозрачный фон.



Экспорт набора фотошаблонов в формате GDSII

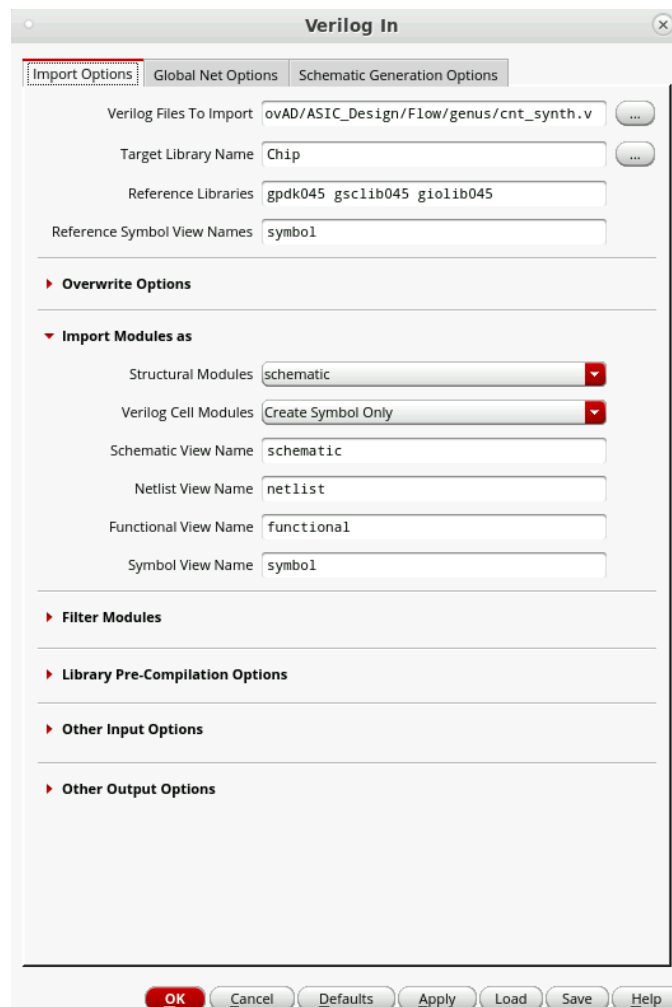
Аналоговое моделирование кристалла

После формирования топологии в потоке проектирования наступает этап верификации схемы на транзисторном уровне. Экстракция паразитных параметров, аналоговое моделирование схемы. Для того, что этот этап произвести необходимо сформировать schematic- и symbol-представление схемы. Для этого следует экспортировать Verilog-описание списка соединений без контактных площадок (можно и с ними).

1 Откройте Virtuoso, и создайте новую библиотеку для Вашего проекта **File > New > Library**. При создании библиотеки выберите Attach to an existing technology library и укажите GPDK045. Или используйте имеющуюся библиотеку.

2 Импортируйте список соединений, для этого в окне CIW выберите **File > Import > Verilog**. На вкладке Import Options укажите v-файл списка

соединений. В поле Target Library Name укажите библиотеку из п. 1, в которую будет импортироваться файл. В поле Reference Libraries укажите библиотеки gpdk045, gsc1ib045 и giolib045. Нажмите ОК. Пройдёт процесс импорта, откроется журнал. В библиотеке будут созданы ячейки с модулями и schematic- и symbol-представлениями.



3 По опыту, полученному на практических занятиях, проведите моделирование схемы без паразитных параметров.

1.a По опыту, полученному на практических занятиях, проведите моделирование схемы после экстракции паразитных параметров.

ПРИЛОЖЕНИЕ А

Листинг А.1 – Описание tcl-скрипта для Genus

```
#Указание выходной директории
set OUTPUT_DIR ./run_dir
#Установка пути к библиотекам
set_attribute lib_search_path ../library
#Указание библиотек для поиска в папке library
set_attribute library {gsclib045_fast.lib}
set_attribute lef_library {\
./lef/gsclib045_tech.lef \
./lef/gsclib045_macro.lef \
./lef/giolib045.lef}

#загрузка HDL-файла под названием file.v
load -v2001 ../rtl/file.v
set DESIGN file
#Подготовка дизайна
elaborate
#Синтез на примитивах
syn_gen
#Сопоставление примитивов и целевой библиотеки
syn_map
#запись синтезированного файла
write -mapped > ${DESIGN}_synth.v
#Запись файла с временными ограничениями
write_sdc > ${DESIGN}.sdc
#Запись дизайна для Innovus
write_design -innovus ${DESIGN}

exit
```


ПРИЛОЖЕНИЕ Б

Листинг Б.1 – Tcl-скрипт для полуавтоматизированной разводки кристалла в
Innovus

```
#-----Предварительная настройка дизайна-----#
set FILE_NAME genus_invs_des/genus.v
set TOP_CELL top
set CORE_SIZE 20
set CORE_MARGIN_IO 30

#Инициализация файла Verilog-описания схемы
set init_verilog $FILE_NAME
#Инициализация ячейки верхнего уровня
set init_top_cell $TOP_CELL
#Инициализация файла входов/выходов
set init_io_file top.save.io
#Инициализация пути к mmmc-файлу
set init_mmmc_file genus_invs_des/genus.mmode.tcl
#Инициализация пути к lef-файлам
set init_lef_file {\
./lib/lef/gscplib045_tech.lef \
./lib/lef/gscplib045_macro.lef \
./lib/lef/giolib045.lef}
#Инициализация проводов питания
set init_pwr_net {VDDO VDDC}
set init_gnd_net {VSSO VSSC}

#-----Инициализация дизайна-----#
#Инициализация проекта
init_design

#Высота/Ширина Использование_ядра Расстояние_до_границы_ядра_4_шт
#floorPlan -site CoreSite -r 0.9975 0.012643 30.0 30.0 30.0 30.0
#Ширина Высота Расстояния_до_границы_ядра_4_шт
floorPlan -site CoreSite -s $CORE_SIZE $CORE_SIZE $CORE_MARGIN_IO
$CORE_MARGIN_IO $CORE_MARGIN_IO $CORE_MARGIN_IO

#Добавление филлеров к контактным площадкам (west, north, south, east)
addloFiller -cell padIORINGFEED10 -prefix FILLER -side w
addloFiller -cell padIORINGFEED10 -prefix FILLER -side s
addloFiller -cell padIORINGFEED1 -prefix FILLER -side w
addloFiller -cell padIORINGFEED1 -prefix FILLER -side s

#Подключение глобальных сигналов питания к соответствующим пинам
clearGlobalNets
globalNetConnect VDDC -type pggpin -pin VDD -instanceBaseName *
globalNetConnect VSSC -type pggpin -pin VSS -instanceBaseName *
```

```
globalNetConnect VDDO -type pggpin -pin VDDIOR -instanceBaseName *
globalNetConnect VSSO -type pggpin -pin VSSIOR -instanceBaseName *
```

```
#Добавление колец питания вокруг ядра
```

```
addRing -nets {VDDC VSSC} -type core_rings -follow io -layer {top Metal3 bottom
Metal3 left Metal2 right Metal2} -width {top 8 bottom 8 left 8 right 8} -spacing {top
1.25 bottom 1.25 left 1.25 right 1.25} -center 1
```

```
#Добавление рельс питания к стандартным ячейкам
```

```
setSrouteMode -viaConnectToShape {noshape}
sroute -connect { blockPin padPin padRing corePin floatingStripe } -layerChangeRange
{ Metal1(1) Metal11(11) } -blockPinTarget { nearestTarget } -padPinPortConnect
{ allPort oneGeom } -padPinTarget { nearestTarget } -corePinTarget { firstAfterRowEnd
} -floatingStripeTarget { blockring padring ring stripe ringpin blockpin followpin } -
allowJogging 1 -crossoverViaLayerRange { Metal1(1) Metal11(11) } -nets { VDDC
VSSC } -allowLayerChange 1 -blockPin useLef -targetViaLayerRange { Metal1(1)
Metal11(11) }
```

```
#Размещение стандартных ячеек с игнорированием скановых цепей
```

```
setPlaceMode -fp false
setPlaceMode -place_global_exp_allow_missing_scan_chain true
place_design
```

```
#Оптимизация разводки
```

```
setNanoRouteMode -quiet -routeTopRoutingLayer default
setNanoRouteMode -quiet -routeBottomRoutingLayer default
setNanoRouteMode -quiet -drouteEndIteration default
setNanoRouteMode -quiet -routeWithTimingDriven false
setNanoRouteMode -quiet -routeWithSiDriven false
routeDesign -globalDetail
```

```
#Добавление филлеров в ядро
```

```
addFiller -cell FILL64 FILL32 FILL16 FILL8 FILL4 FILL2 FILL1 -prefix FILLER -markFixed
```

```
#Экспорт DEF-файла
```

```
set dbgLefDefOutVersion 5.8
global dbgLefDefOutVersion
defOut -floorplan -netlist -routing top.def
```

```
#Сохранение дизайна
```

```
saveDesign top.enc
```

```
exit
```

Часто возникающие вопросы

В: В папке с проектом у меня накопилось много файлов с расширением cmd и log. Для чего они нужны и можно ли их удалить?

О: Это файлы, которые автоматически генерируются Genus и Innovus. В cmd сохраняется информация о действиях (какие кнопки нажимались и какие параметры выставлялись). Этот файл удобно использовать для дальнейшего формирования скрипта на языке Tcl. В log сохраняется информация о результатах действий, он выступает в роли журнала. Эти файлы генерируются автоматически каждую сессию работы Genus и Innovus. Поэтому они не нужны и их можно смело удалить. Быстро удалить эти файлы можно командой в терминале Linux, запущенном в данной папке:

```
rm *.cmd* *.log*
```

В: На этапе размещения стандартных ячеек после нажатия ОК что-то происходит, окно закрывается, но ячеек на топологии нет. Что это может быть?

О: Если в терминале в конце в качестве отчёта указано, что ошибок нет, но есть предупреждения, то может быть следующее:

1 Если в качестве предупреждения написано, что нет размещаемых ячеек, то скорее всего в файле genus.v в модуле верхнего уровня какой-то инстанцируемый модуль имеет ошибку в названии. Проверьте все названия модулей и если это действительно так, то укажите верное название и заново начните формирование топологии.

2 Если в качестве предупреждения указано, что какой-то пин не может быть соединён с каким-либо сигналом, то в файле genus.v проверьте, чтобы к контактной площадке не была подключена шина. Помните, что к контактной площадке можно подключать только одиночные сигналы (физически – один провод), а не их набор!

В: На этапе размещения стандартных ячеек после нажатия ОК что-то происходит, окно не закрывается и ячеек на топологии нет. Что это может быть?

О: Если в терминале есть указание на ошибку, связанную с появлением в дизайне скановых цепей, то необходимо сделать следующее. Скановые цепи формируются тогда, когда появляются триггеры с разрешающим входом enable. В окне терминала введите команду:

```
> setPlaceMode -place_global_exp_allow_missing_scan_chain
true
```

Затем снова нажмите на размещение дизайна.

В: Так какая должна быть иерархия папок в проекте и где должен лежать файл cds.lib?

О: В проекте важна упорядоченность, для этого удобно сформировать следующее дерево:

```
/Project
../rtl
....<file>.v
../syn
..../genus_invs_des
.....genus.v
.....genus.globals
.....genus.mmode.tcl
../library
....gsclib045_fast.v
....gsclib045_fast.lib
..../lef
.....gsclib045_tech.lef
.....gsclib045_macro.lef
.....giolib045.lef
../virtuoso
....cds.lib
```

В: При разводке кристалла у меня не подключаются контактные площадки питания ядра к кольцам VDDC и VSSC, что делать?

О: Ничего.