Министерство науки и высшего образования Российской Федерации

Томский государственный университет систем
управления и радиоэлектроники

А.А. Бомбизов

СИНТЕЗ ИНТЕГРАЛЬНЫХ СХЕМ

Методические указания к выполнению
лабораторной и самостоятельной работы
по дисциплине «Проектирование систем на кристалле»

Томск
2022

УДК 004.42
ББК 32.973
Б 803

Рецензент:
Тренкаль Е.И., доцент кафедры конструирования узлов и деталей радиоэлектронной аппаратуры ТУСУР, канд. техн. наук

Бомбизов, Александр Александрович
Б 803 Синтез интегральных схем: методические указания к выполнению лабораторной и самостоятельной работы по дисциплине «Проектирование систем на кристалле» / А.А. Бомбизов. – Томск. гос. ун-т систем упр. и радиоэлектроники, 2022. – 17 с.

Методическое пособие содержит описание и порядок выполнения лабораторной работы по теме «синтез интегральных схем» и содержание разделов отчета.

Одобрено на заседании каф. КУДР, протокол № 234 от 5 марта 2022 г.

УДК 004.42
ББК 32.973

1 Введение

Логический синтез в электронике – процесс получения списка соединений логических вентилей из абстрактной модели поведения логической схемы (например, на уровне регистровых передач). Наиболее распространенный пример этого процесса – синтез спецификаций, написанных на языках описания аппаратуры. Синтез выполняют программы-синтезаторы, способные оптимизировать проект согласно различным особенностям устройства, таким как временные ограничения, площадь и используемые компоненты. Такие программы обычно специализируются на генерации битовых потоков для программируемой логики или создании интегральных схем специального назначения. Логический синтез является составной частью автоматизации проектирования электронных приборов.[1]

Данные методические указания описывают логический синтез verilog-описания в verilog-представление в виде комбинационной схемы из библиотечных компонентов. Дополнительно здесь выполняется построение временных диаграмм распространения сигнала с учетом заявленных библиотечных задержек.

Материал предназначен для магистрантов, изучающих дисциплину проектирование систем на кристалле.

Работа должна быть выполнена в принятом для технических работ виде и оформлена согласно образовательному стандарту вуза ОС ТУСУР 01-2021 или более поздней редакции.

2 Исходные данные

Для начала работы потребуется:

1. Verilog-описание целевого модуля синтеза, отладка которого уже должна быть выполнена в ncverilog и simvision.

2. Cadence IC 6.1.5.

3. Библиотека gsclib045 – учебная библиотека GSCLib Cadence с технологической нормой 45 нм. Архив с библиотекой gsclib045_all_v4.4.tgz необходимо скачать по ссылке в SDO и распаковать в каталог exchange.

3 Порядок логического синтеза

1. Порядок синтеза будет рассмотрен на примере verilog-описания модуля мультиплексора Mux.v;

[1] Логический синтез: Материал из Википедии – свободной энциклопедии. URL: https://ru.wikipedia.org/wiki/Логический_синтез (Дата обращения 01.07.2022)

2. Создайте каталог проекта, например, prj;

3. Скопируйте в созданный каталог файл с описанием, например, MUX.v и его тестовый стенд для функционального анализа tbMUX.v;

4. Откройте терминал и перейдите в каталог prj;

5. Запустите программу Encounter RTL Compiler для выполнения синтеза командой *rc -gui* (флаг *gui* не является обязательным), после чего должны появится графический интерфейс и приглашение ко вводу *rc:/>* как показано на рисунке 1;

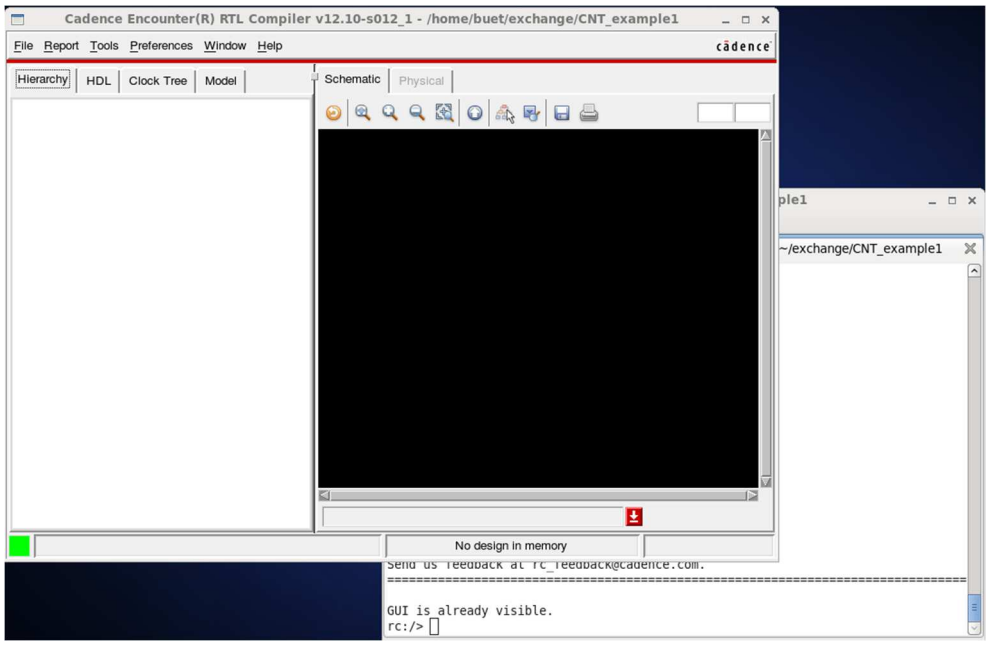


Рисунок 1 – Результат выполнения команды rc -gui

6. После запуска появится приглашение ввода Tcl-команд (Tcl (от англ. Tool Command Language – «командный язык инструментов», читается «тикль» или «ти-си-эль») – скриптовый язык высокого уровня). В приглашении ввода можно набрать команду *ls* для просмотра созданного виртуального окружения проекта;

7. Подайте команду чтения verilog-описания: *read_hdl -v2001* MUX.*v*, где флаг *-v2001* означает язык читаемого файла verilog 2001, а MUX.*v* сам файл.

8. Далее для RTL-синтеза необходимо подключить библиотеку с задержками распространения сигналов в логических элементах. Для этого должен быть взят один из файлов набора библиотеки gsclib045 по следующему пути gsclib045/timing. В указанном каталоге размещены различные наборы, различающиеся по быстродействию синтезируемой топологии и напряжению питания. Для решения задачи синтеза будет выбрана библиотека элементов *slow_vdd1v0_basicCells.lib*;

9. Для подключения выбранной библиотеки выполните следующую команду: *set_attribute library /home/buet/exchange/gsclib045_all_v_4.4/ gsclib045/timing/slow_vdd1v0_basicCells.lib*, где параметр для атрибута *library* является путь до библиотеки;

10. Далее необходимо подключить технологические библиотеки, в которых указаны параметры слоев микросхемы. Они размещены в каталоге *gsclib045/lef*.

11. Для подключения необходимо выполнить следующую команду

set_attribute lef_library
{/home/buet/exchange/gsclib045_all_v4.4/GSCLIB045/lef/ gsclib045_macro.lef
/home/buet/exchange/gsclib045_all_v4.4/GSCLIB045/lef/
gsclib045_multibitsDFF.lef
/home/buet/exchange/gsclib045_all_v4.4/GSCLIB045/lef/ gsclib045_tech.lef}

Обратите внимание, что для подключения сразу нескольких библиотек их нужно объединить фигурными скобками разделяя пробелами.

12. Далее выполните RTL-синтез путём вызова команды *elaborate,* по результатом которой будет выведена информация, как изображено на рисунке 2.

```
Info    : Elaborating Design. [ELAB-1]
        : Elaborating top-level block 'MUX' from file 'MUX.v'.
Info    : Done Elaborating Design. [ELAB-3]
        : Done elaborating 'MUX'.
/designs/MUX
rc:/>
```

Рисунок 2 – Результат выполнения команды elaborate

В результате в виртуальном каталоге /designs будет создан каталог */MUX*, содержимое которого можно просмотреть, выполнив команду *ls /designs/MUX*;

13. Далее, если используется оптимизация по быстродействию, то нужно задать ограничение созданием тактового сигнала, чтобы синтезатор во время своей работы при генерации кристалла попытался выполнить маршрут таким образом, чтобы максимальное время прохождения сигнала укладывалось в период выбранного тактового сигнала.

define_clock -name C -period 10000 CLK

где 10000 – время в пикосекундах;

CLK – порт тактового сигнала, который определен в описании модуля;

14. Перед следующей командой необходимо отключить пользовательский интерфейс с помощью команды *gui_hide*, иначе синтезатор зависнет;

15. Оптимизация и синтез из стандартных ячеек выполняется с помощью команды *synthesize*. В результате должно появиться сообщение *Synthesis succeeded*;

16. Отобразить пользовательский интерфейс для просмотра результата синтеза можно с помощью команды *gui_show*;

17. Команда *report area* отобразит текущую занимающую площадь, а *report timing* – время прохождения сигнала (*Timing slack*). Аналогичные отчеты можно получить в пользовательском интерфейсе;

18. Постоптимизация с попыткой уменьшить дистанции цепей и др. выполняется командой *synthesize -to_mapped.* В пользовательском интерфейсе можно посмотреть результат оптимизации и замены стандартных элементов на библиотечные;

19. Далее результат синтеза необходимо сохранить в файл путем вызова команды:

write_hdl -v2001 > MUX_syn.v

где *> MUX_syn.v* выполняет перенаправление вывода стандартного потока в указанный файл *MUX_syn.v*;

20. Дополнительно нужно сохранить маршруты с задержками прохождения сигнала от входов к выходу путем выполнения команды:

write_sdf > MUX_syn.sdf

где *MUX_syn.sdf* – файл-контейнер.

21. Выход из RC выполняется командой exit;

22. Все введенные в Encounter RTL Compiler команды были сохранены в rc.cmd, поэтому повторное использование может значительно ускорить процесс;

23. Далее необходимо проверить результат синтеза путём запуска симуляции синтезированного из библиотечных элементов модуля при помощи следующей команды:

ncverilog +gui +access+rwc tbMUX.v MUX_syn.v -v /home/buet/exchange/gsclib045_all_v4.4/GSCLIB045/verilog/slow_vdd1v0_basicCells.v

где *slow_vff1v0_basicCells.v* – библиотека базовых вентилей, из которых был синтезирован модуль MUX;

В результате в SimVision подгрузятся не только тестовый стенд с тестируемым модулем, но и нужный набор элементов как показано на рисунке 3.

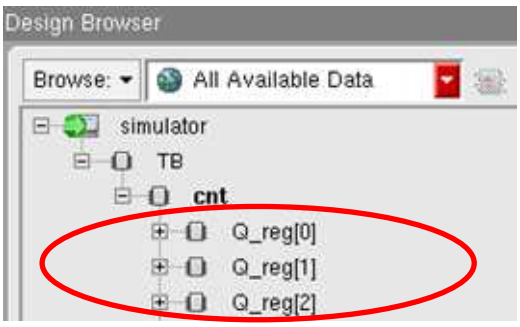


Рисунок 3 – Подгрузка библиотек

Если выбрать синтезированный модуль MUX и нажать в SimVision на кнопку  , то откроется окно с изображением мультиплексора, как показано на рисунке 4. Раскрытие иерархии выполняется двойным нажатием мыши по блоку. Как видно из рисунка, 8-канальный мультиплексор построен из двух 4-канальных мультиплексоров и объединяющем 2-канальном. В свою очередь мультиплексоры состоят из типовых библиотечных вентилей.

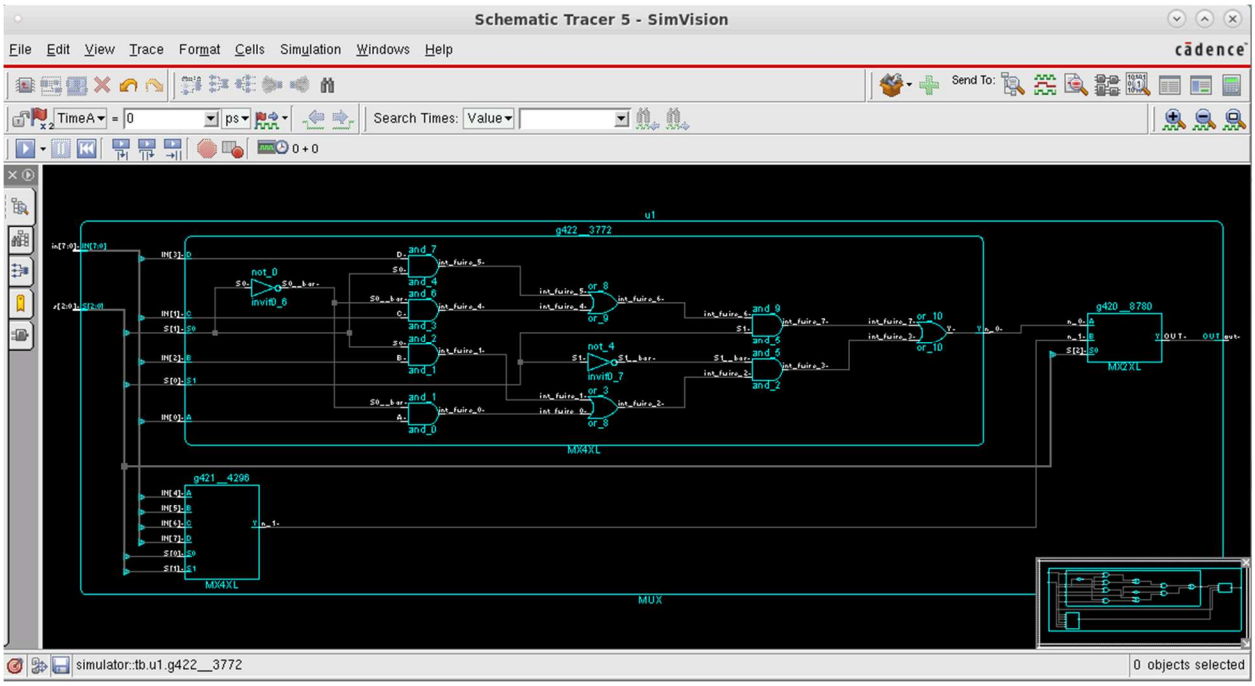


Рисунок 4 – Синтезированный модуль

24. Выполните симуляцию, нажав на кнопку  .

4 Порядок синтеза топологии

1. Для генерации необходимой для синтеза топологии в Encounter RTL Compiler (rc) после команды *write_sdf > MUX_syn.sdf* необходимо выполнить следующую команду:

write_encounter design

2. Выход из Encounter RTL Compiler выполняется командой *exit*;

3. Предварительно нужно подготовить файлы проекта к синтезу топологии. Для этого требуется открыть в каталоге rc_enc_des файл rc.conf и добавить в него следующие строки:

set rda_Input(ui_pwrnet) {VDD}

set rda_Input(ui_gndnet) {VSS}

4. Загрузка в innovus синтезированного на предыдущих этапах списка соединений выполняется при помощи команды:

encounter -init rc_enc_des/rc.enc_setup.tcl

Эта команда подгрузит в проект ранее добавленные библиотеки и сгенерирует прообраз будущего кристалла (рисунок 5).

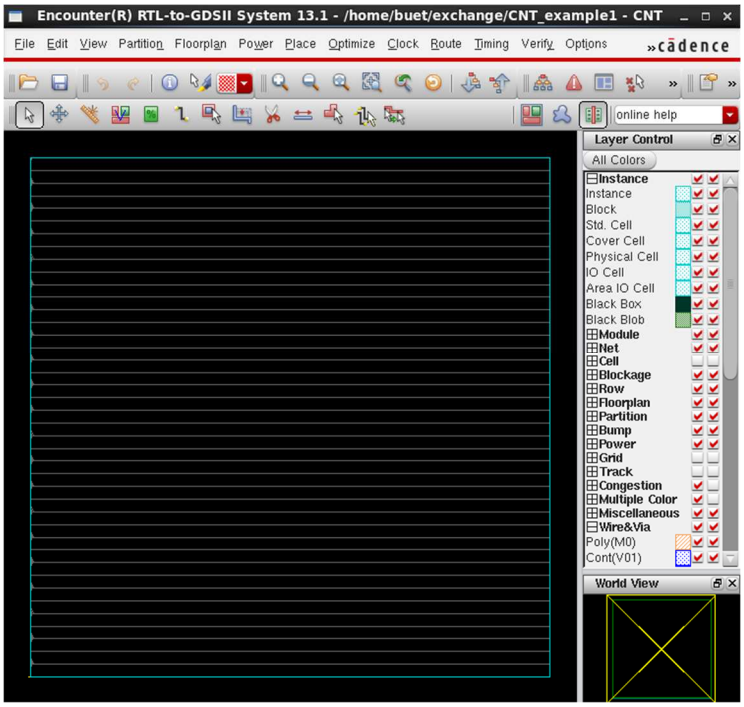
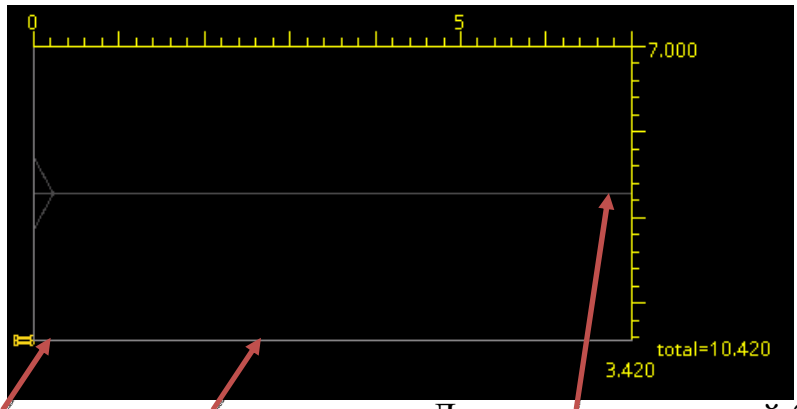


Рисунок 5 – Прообраз кристалла

В результате должна появиться рабочая область кристалла, на которой должны быть размещены элементы как показано на рисунке 6. Измерить размеры (в мкм) можно с использованием горячей клавиши «K» или путем нажатия на кнопку  на панели инструментов. Shift+K убирает все измерения.



Порты модуля — Граница кристалла — Линия, вдоль которой будут укладываться стандартные ячейки

Рисунок 6 – Рабочая область кристалла

5. Далее необходимо выбрать пропорции кристалла и размер области для размещения стандартных ячеек. Для этого необходимо выбрать пункт меню

Floorplan->Specify Floorplan (рисунок 7).

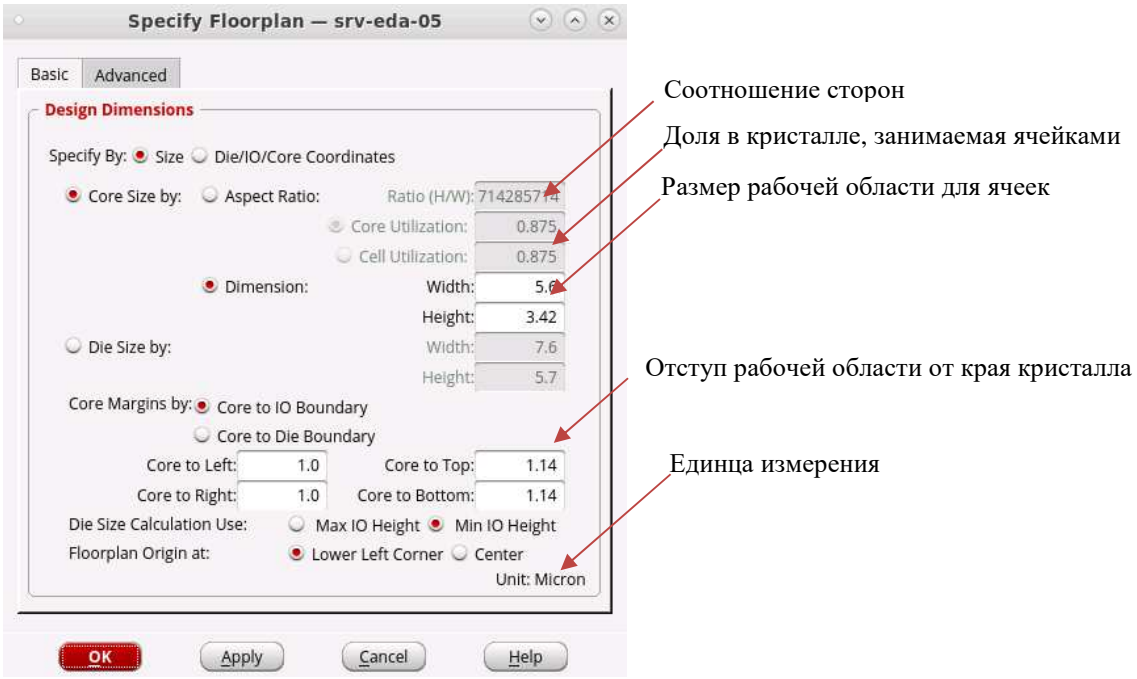


Рисунок 7 – Выбор рабочей области

Отступ рабочей области от края кристалла необходим для размещения колец питания (об этом позже).

6. Размещаются ячейки путем выбора пункта меню

Place->Place Standard Cell…

и в появившемся диалоге нажатия на OK. В результате будут размещены ячейки и выполнена трассировка.

7. В первом приближении нужно убрать трассировку

Клавиша D -> delete -> Objects: All (рисунок 8).

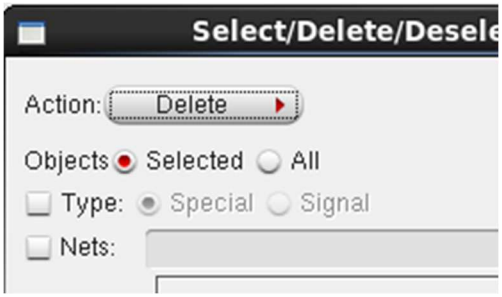


Рисунок 8

В первом приближении должна получиться следующая картина (рисунок 9).

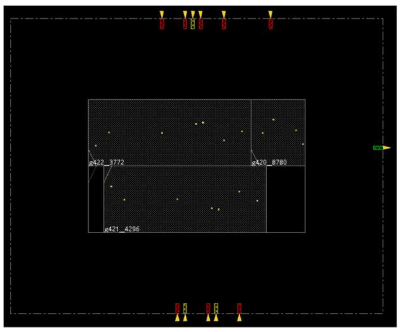


Рисунок 9

У каждой ячейки с одной горизонтали полоска питания, со второй земли. Поэтому они соединяются по строкам, таким образом, чтобы у них совпадала одна из цепей питания.

8. Далее необходимо подключить цепи питания к питанию стандартных ячеек путем выбора пункта меню Power->Connect Global Nets (рисунок 10).

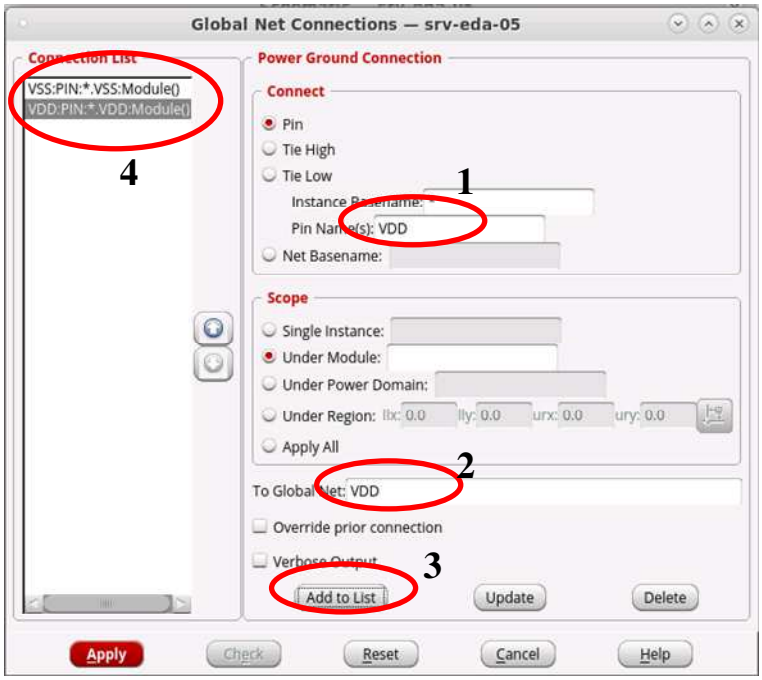


Рисунок 10

Цепи питания, указанные в п. 3, приняты VDD и VSS, они совпадают с соответствующими именами контактов питания стандартных ячеек.

9. Для распределения цепей питания по кристаллу в первую очередь необходимо создать кольца питания путем выбора пункта меню Power->Power Planning->Add Ring (рисунок 11).

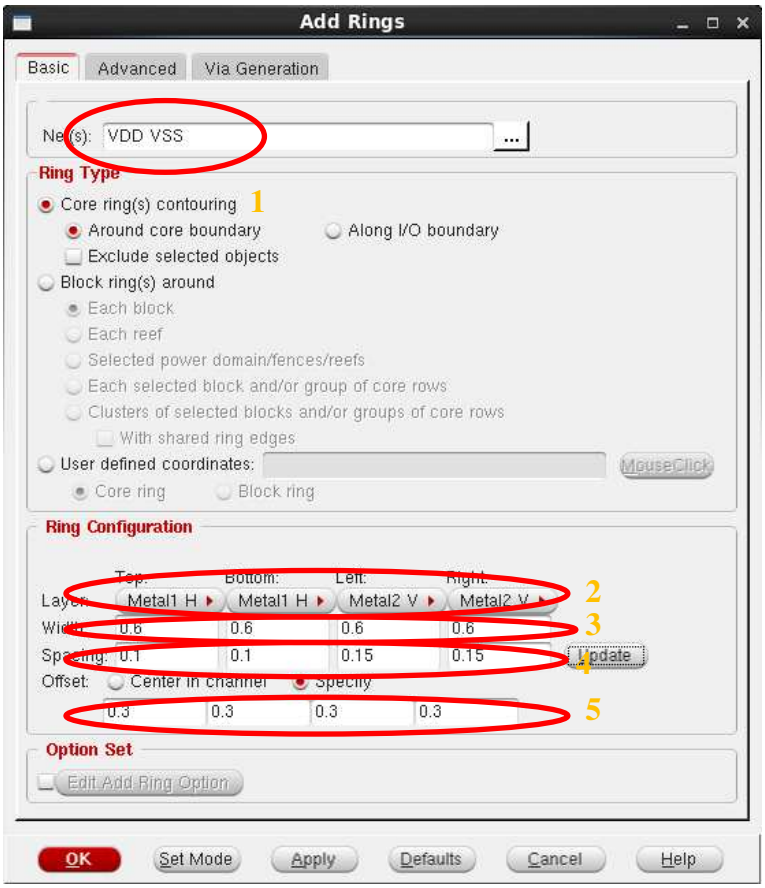


Рисунок 11

В появившемся окне указать

1) для каких цепей создать кольца;

2) слой металлизации для горизонтальной и вертикальной части кольца;

3) ширину кольца (высчитывается исходя из отступов, указанных на рисунке 7);

4) расстояние между кольцами;

5) расстояние от рабочей области.

В результате должны появиться кольца как показано на рисунке 12 (цвет зависит от выбранного слоя).

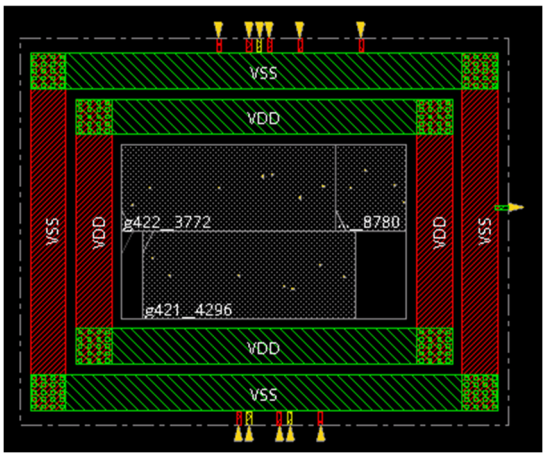


Рисунок 12

10. Соединение ячеек с кольцами

Меню Route->Special Route (рисунок 13).

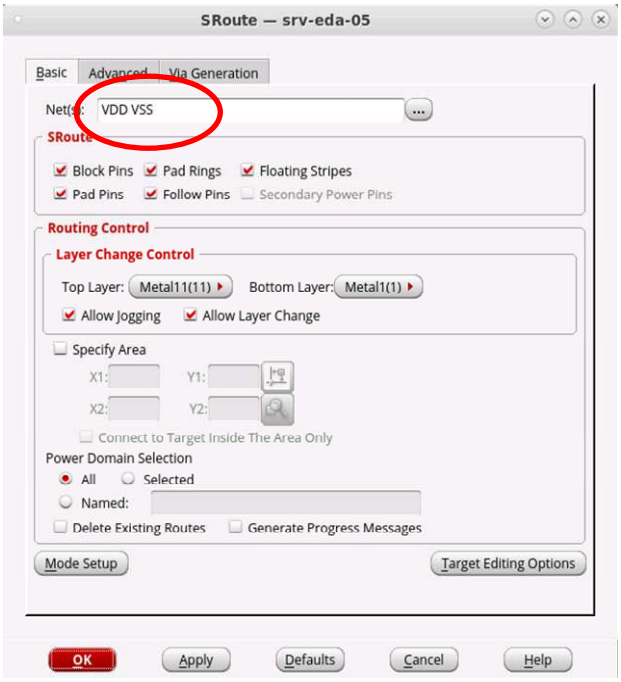


Рисунок 13

Затем ОК и отслеживаем как кольца соединились с линиями питания ячеек.

11. Добавление дополнительных шин питания (если длинные маршруты или вытянутый кристалл) выполняется с использованием следующей операции

Power -> Power Planning -> Add Stripe (рисунок 14).

Рисунок 14

12. Завершаем трассировку путем выполнения команды Меню Route->NanoRoute->Route->Ok (рисунок 15).

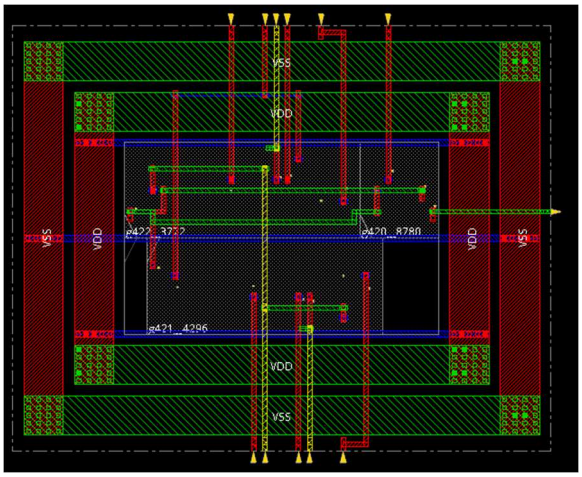


Рисунок 15

13. Далее заполняются пустые места:

Меню Place->Physical Cell->Add filler->Select (рисунок 16, по кнопке Select выбирается типоразмер заглушки).

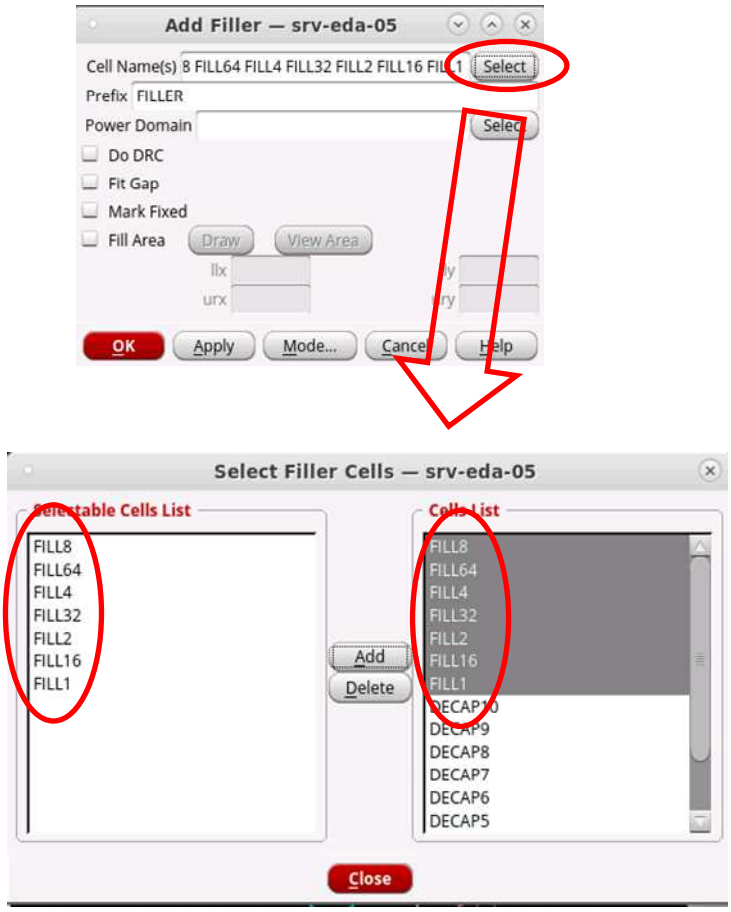


Рисунок 16

14. Далее сохраняем дизайн

File ->SaveDesign

Data Type: Encounter

15. Сохраняем размещение элементов

File->Save->DEF…: MUX.def

16. Далее сохраняется список соединение

File->Save->Netlist: имя MUX_layout.v

17. Далее из полученной топологии извлекаются задержки

Timing->Extract RC (рисунок 17).

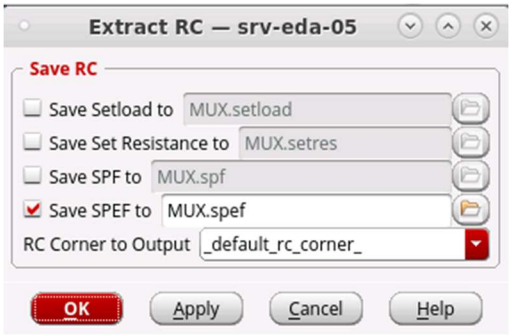


Рисунок 17

18. Экспортируются задержки

Timing -> Write sdf: MUX_layout.sdf

19. Для загрузки Restore Design и переключить вид в режим  physical view.

20. Для получения выходного файла для производства необходимо выполнить

File->Save->GDS/OASIS (рисунок 18)

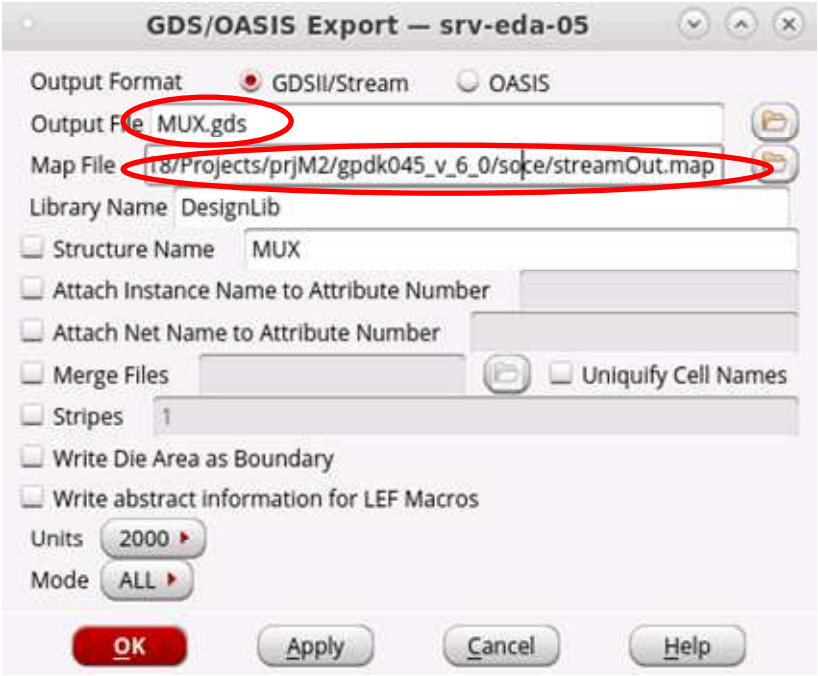


Рисунок 18

Output File – имя сохраняемого файла;

Map File – файл с сопоставлением слоев среды разработки и технологической базы

5 Примерные темы работы

Темы работ определяются преподавателем кафедры. Ниже приведен примерный перечень тем работ:

1) Загружаемый по последовательному синхронному интерфейсу 32-битный счетчик (загрузка начального состояния, загрузка коэффициента счета);

2) Преобразователь из параллельного интерфейса в последовательный с линией синхронизации. В конце каждой выходной посылки должен быть установлен бит четности;

Конфигурируемый передатчик RS-232

№	Частота тактирования, МГц	Обеспечиваемые скорости, кбит/с	Бит четности	Размер пакета
3	8	9600,19200	Even	7
4	16	57600, 115200	Odd	6

Конфигурируемый приемник RS-232

№	Частота тактирования, МГц	Обеспечиваемые скорости, кбит/с	Бит четности	Размер пакета
5	8	9600,19200	Even	7
6	16	57600, 115200	Odd	6

7) Сумматор, загружаемый по последовательному порту, выгружаемый по параллельному (аргументы 8 бит, результат 9);

8) Умножитель, загружаемый по параллельному и выгружаемый по последовательному порту (аргументы 8 бит, результат 16);

9) Устройство возведения в целочисленную степень, загружаемое по параллельному и выгружаемое по последовательному порту (основание 4 бита, показатель 4 бита, результат 60 бит);

10) SPI-передатчик с возможностью выбора полярности и фазы сигнала;

11) SPI-приёмник с возможностью выбора полярности и фазы сигнала;

12) Кодировщик кодом Баркера с возможностью выбора последовательности;

13) Декодировщик кодом Баркера с возможностью выбора последовательности;

6 Порядок выполнения и структура отчета

Выполнение работы должно начинаться с ознакомления с темой. Далее должен быть выполнен литературный обзор по тематически связанным ресурсам, собрана техническая документация на используемые технические средства. Далее должна быть описана и согласована с преподавателем логика работы разрабатываемого модуля в соответствии с выбранным вариантом задания. На следующем этапе должен быть реализован в программном коде (на языке verilog) алгоритм работы модуля.

Далее должен быть выполнен логический синтез и синтез топологии. После тестирования RTL-описания должен быть подведен итог, написан

технический отчет и выполнена защита работы с демонстрацией работы симулятора и топологии преподавателю.

Примерная структура отчёта должна быть следующая:

1) титульный лист, оформленный в соответствии с ОС ТУСУР 01-2021 или более поздней редакцией;

2) индивидуальное задание;

3) оглавление;

4) введение. Должно содержать тему и актуальность выполняемой работы;

5) анализ индивидуального задания, включающий в себя литературный обзор по тематически связанным ресурсам, сбор технической документации;

6) описание структурной схемы, логики и алгоритма (сценария) разрабатываемого модуля;

7) разработка программного обеспечения в соответствии с индивидуальным заданием. Данный раздел должен содержать описание структуры разработанной программы;

8) описание тестирования результатов синтеза verilog-описания;

9) описание тестирования результатов логического синтеза;

10) описание синтеза топологии;

11) описание тестирования результатов синтеза топологии;

12) заключение, содержащее перечень основных полученных в работе результатов и сделанных выводов. В него могут включаться рекомендации относительно перспектив продолжения данной работы;

13) список использованных источников. На все перечисленные в списке;

14) Приложение с листингом программы.