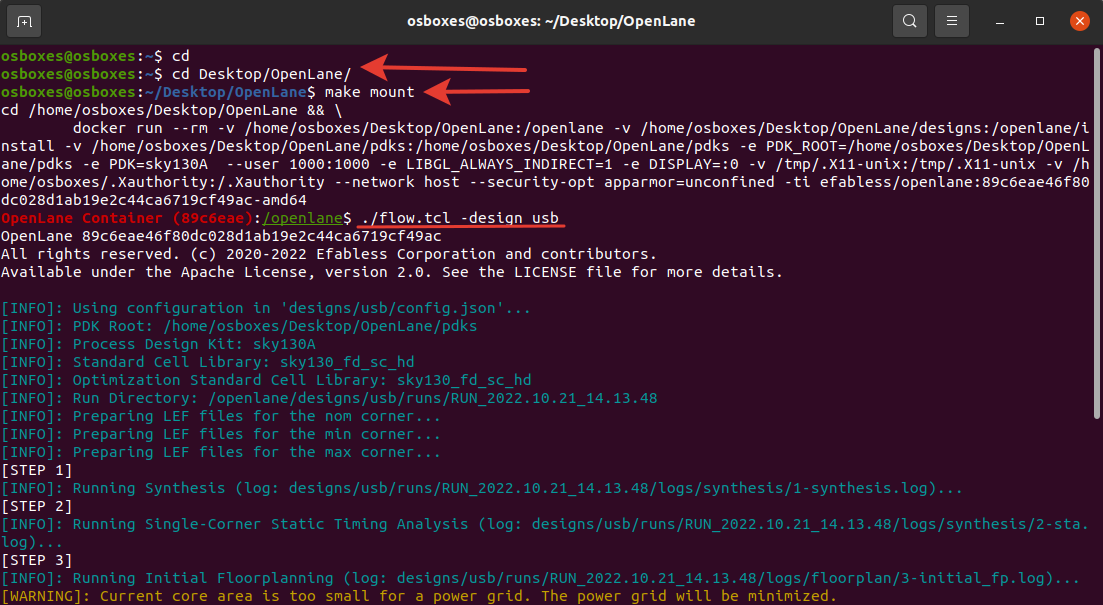
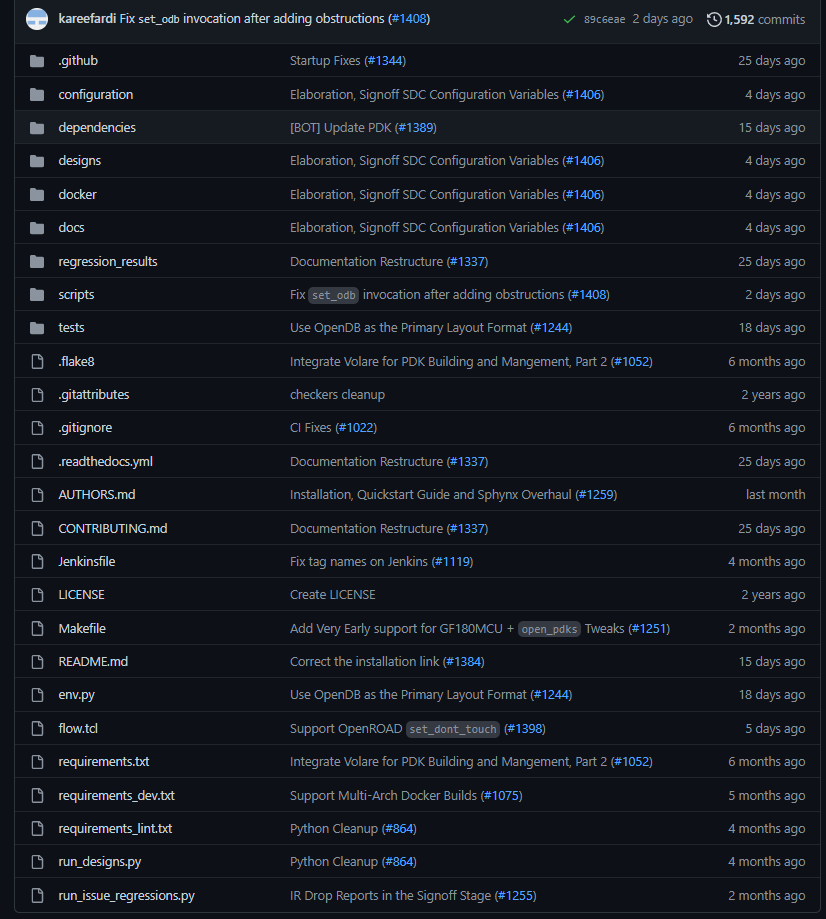
**Вступление OpenLane**

Перед объяснением всего что будет происходить, запустим маршрут, а пока он будет выполняться мы будем разбираться в том что происходит в процессе.  
  
Для примера будем использовать схему из репозитория OpenLane под названием usb  
OpenLane/designs/usb  
  
План действий:  
1) Открыть терминал и перейти в домашнюю директорию  
**cd**  
2) перейти в директорию OpenLane  
**cd /Desktop/OpenLane/**  
3) Запустить OpenLane в докере с помощью  **make mount**  
4)А теперь приступим к запуску flow.tcl  
**./flow.tcl -design usb**

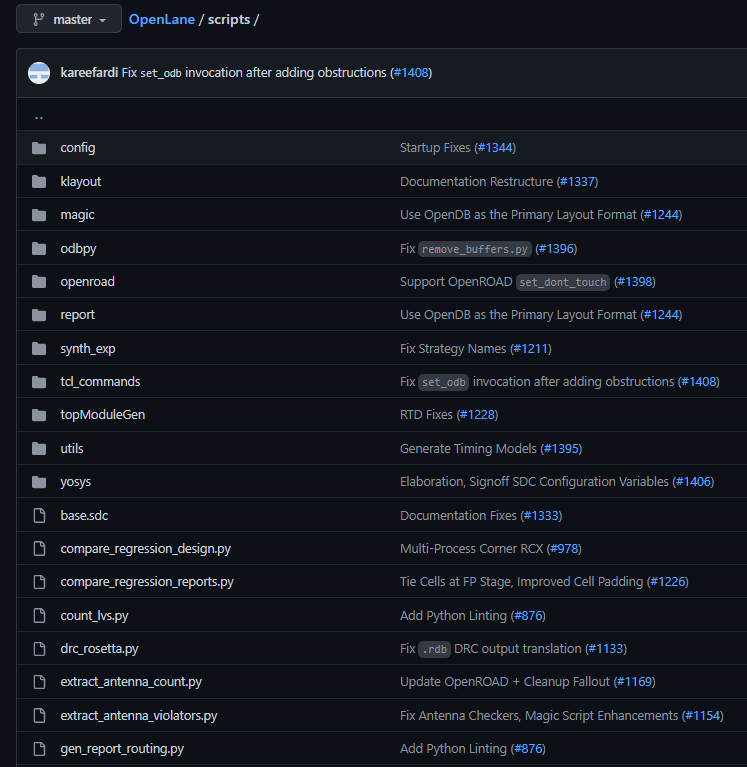


Если упростить, то OpenLane это набор скриптов которые соединяют разные программы для полноценного прохождения маршрута RTL to GDSII такие как OpenROAD, Yosys, Magic, Netgen и другие, а также свои скриптовые методы исследования и оптимизации дизайна.



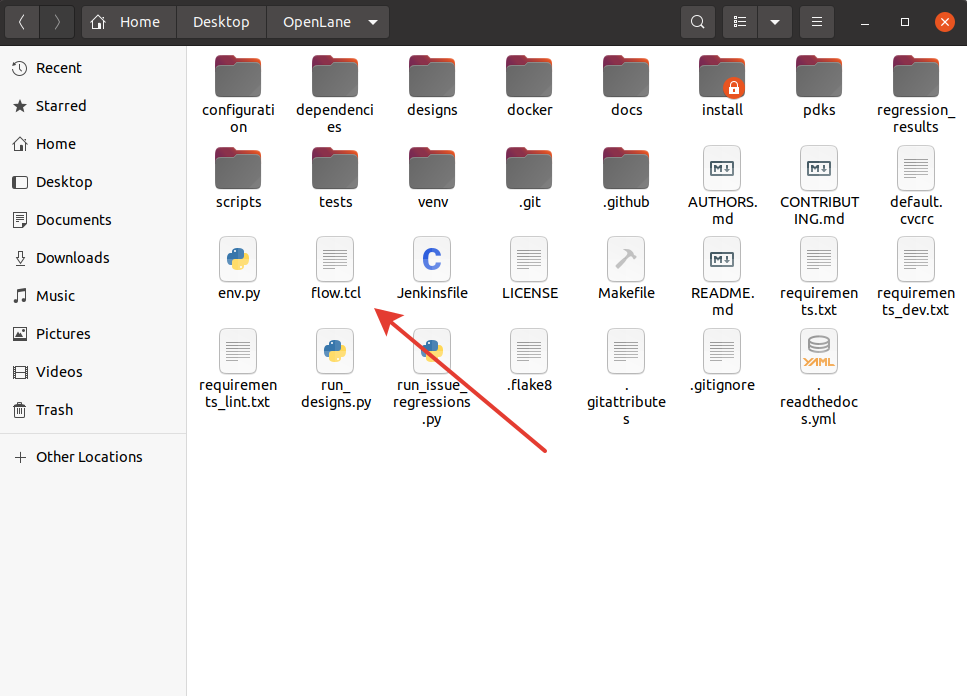
Для начала рассмотрим основные моменты которые нас интересуют из репозитория:

Папка scripts:



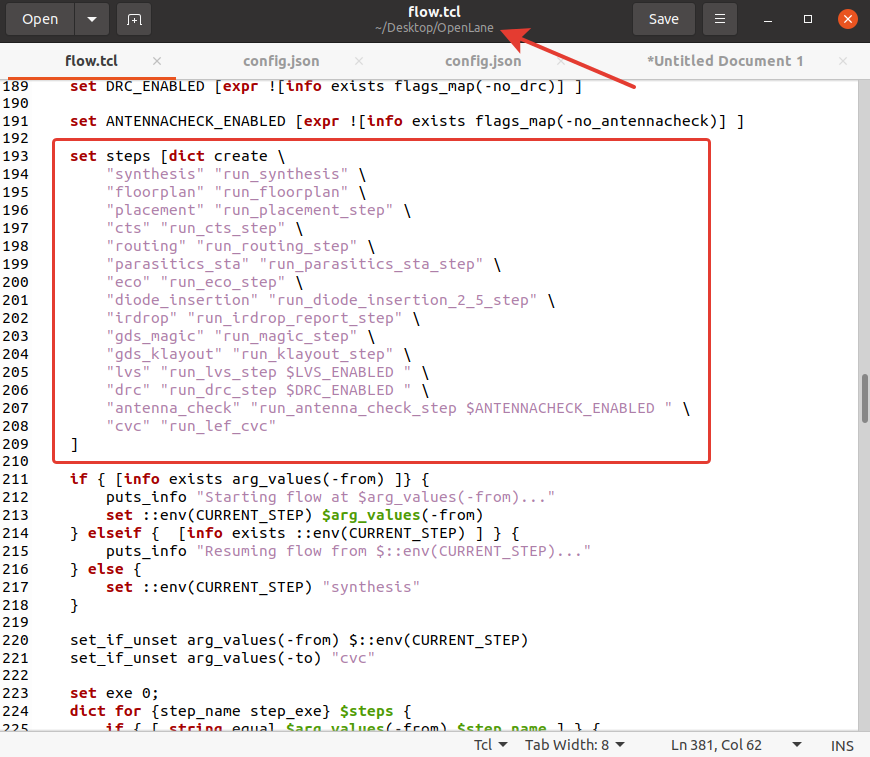
В данной папке scripts как раз находятся «Связи» для взаимодействия этих программ.  
Нам не потребуется вносить какие-либо изменения в эту папку.

В OpenLane в главном разделе есть специальный скрипт на языке TCL “flow.tcl”  
в котором реализованно взаимодействие и вызов всех нужных для прохождение полного маршрута действий в правильном порядке. Нам нужно лишь запустить его указав название схемы, и весь маршрут будет проходить самостоятельно. Как запустить маршрут я расскажу позднее.

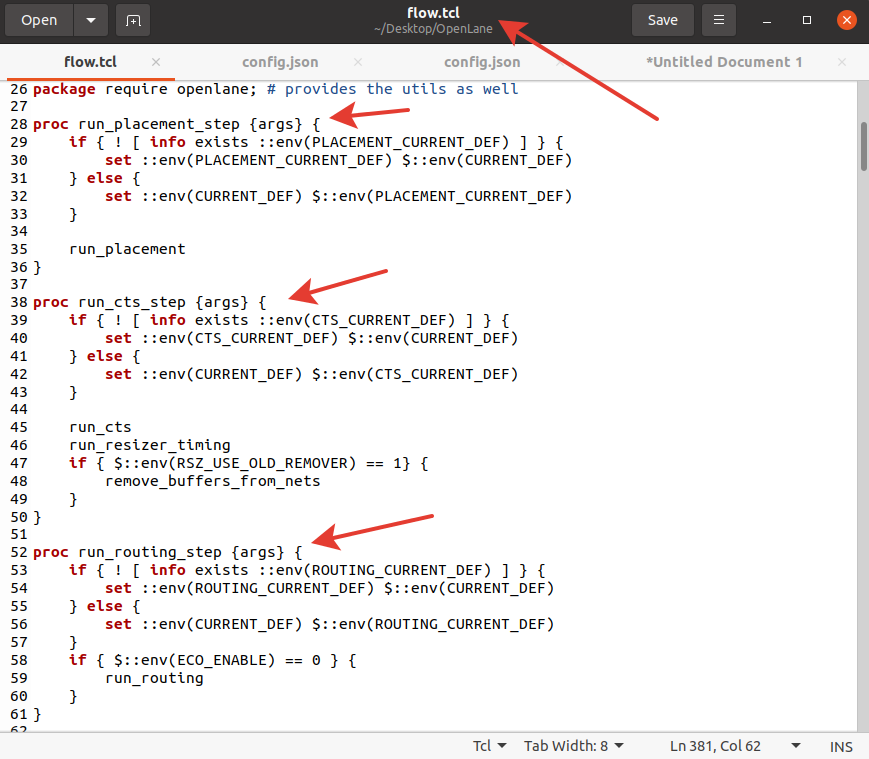


Для начала рассмотрим основные моменты flow.tcl

На данном скриншоте представлены шаги выполнения потока OpenLane:  
  
synthesis  
floorplan  
placement  
и так далее...

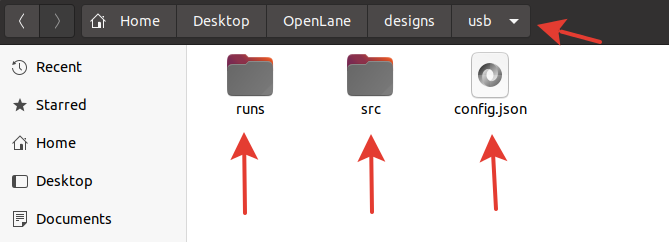


В дальнейшем эти “шаги” запускают скрипты из директории OpenLane/scripts и подтягивают различные параметры.



**Структура папки с дизайном. Файл конфигурации**

Папка с дизайном находится по следующему адресу OpenLane/designs/<Название дизайн>



1) В папке с дизайном есть runs, где находятся все данные о завершенном выполнении маршрута.  
финальные результаты OpenLane/designs/usb/runs/<RUN\_\*\*\*>/results/final  
логи и репорты  
OpenLane/designs/spm/runs/<RUN\_\*\*\*>/logs  
OpenLane/designs/spm/runs/<RUN\_\*\*\*>/reports

2) Помимо папки runs где лежат результаты выполнения маршрута, есть папка src где находится rtl описание дизайна

3) Для того что бы внести корректировки в маршрут(такие как площадь дизайна и тд) необходимо изменять файл конфигурации конкретного дизайн проекта.  
**OpenLane/designs/usb/config.json**  
Документация по всем параметрам конфигураций:  
**https://openlane.readthedocs.io/en/latest/reference/configuration.html**  


**Запуск маршрута**

Рассмотрим 3 варианта запуска маршрута:

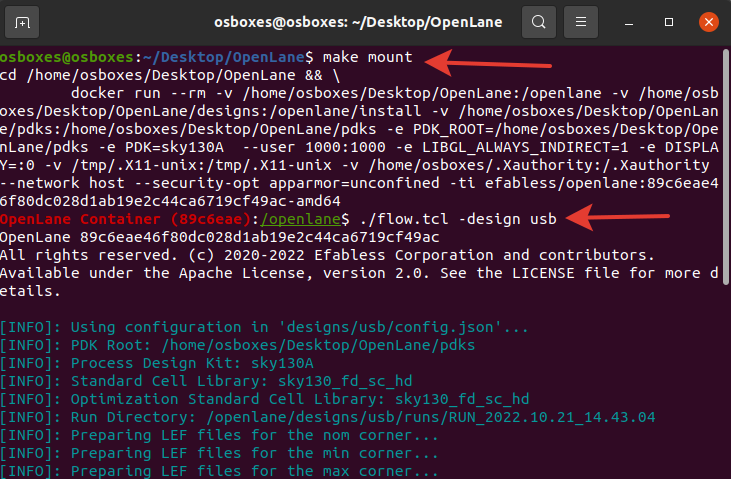
- Обычный автоматический поток  
- Интерактивный режим  
- Регрессионные тесты  
  
**1) Запуск автоматического потока**

OpenLane запускается с помощью Докера.  
Для начала необходимо перейти в директорию OpenLane:

**cd /Desktop/OpenLane**

Теперь нужно запустить OpenLane:

**make mount**



После того как мы запустили OpenLane в контейнере мы можем приступить к практической части.  
Выше мы уже упоминали про flow.tcl теперь рассмотрим как его запускать. Важно запустить flow.tcl именно в контейнере OpenLane.  
  
Пример: **./flow.tcl -design \*Название дизайна\***  
  
Всё. Больше делать ничего не нужно, он сам полностью завершит маршрут. (Либо вывалится из за ошибки)  
  
**2) Интерактивный режим.**

Для начала перейдем в директорию OpenLane если вы ещё не там, а потом запустим OpenLane в контейнере

**cd /Desktop/OpenLane  
make mount**

Помимо автоматического прохождения полного маршрута, можно самим поэтапно запускать все скрипты в flow.tcl.

**./flow.tcl -interactive**

Далее после запуска интерактивного режима нужно выбрать дизайн который будет запускаться:  
**prep -design <design> -tag <tag> -config <config> -init\_design\_config -overwrite**  
Важно указать дизайн, остальные параметры можно не указывать

Пример:  
**prep -design usb  
  
run\_synthesis  
run\_floorplan  
run\_placement  
run\_cts  
run\_routing**

**write\_powered\_verilog** с последующим set\_netlist **$::env(routing\_logs)/$::env(DESIGN\_NAME).powered.v**

**run\_magic  
run\_magic\_spice\_export  
run\_magic\_drc  
run\_lvs  
run\_antenna\_check**

Более подробно про интерактивный режим и команды можно посмотреть в следующих ссылках:  
<https://openlane.readthedocs.io/en/latest/reference/interactive_mode.html>  
<https://openlane.readthedocs.io/en/latest/reference/openlane_commands.html>

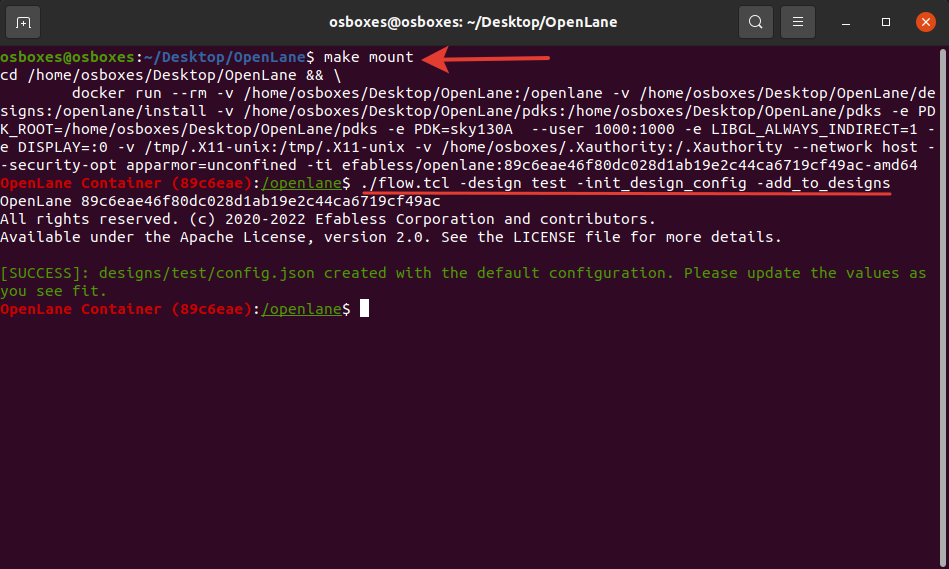
**3) Регрессионный запуск.**  
  
У этого метода есть 2 применения которые можно совмещать:  
- Одновременный запуск маршрута с несколькими дизайнами параллельно  
- Исследование дизайна с различными изменениями в конфигурационных параметрах

Для начала перейдем в директорию OpenLane если вы ещё не там, а потом запустим OpenLane в контейнере

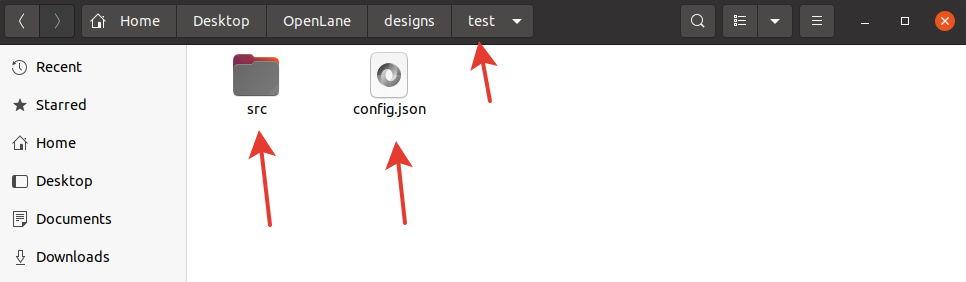
**cd /Desktop/OpenLane  
make mount**Запуск 4-х дизайнов в 4 потока (параллельно): **python3 run\_designs.py --threads 4 spm xtea s44 usb**Запуск 2-х дизайнов в 2 потока с использованием специального конфигурационного файла  
**python3 run\_designs.py --regression ./scripts/config/regression.config --threads 2 spm xtea**  
Данные параметры конфигурационного файла добавляются к основному который лежит в папке с дизайном

**Добавление своего дизайна**

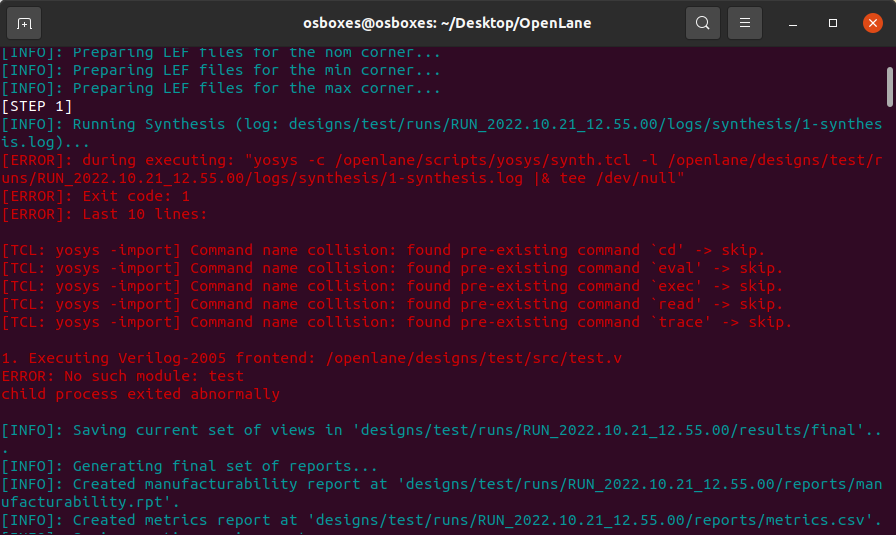
Для того чтобы добавить свой дизайн достаточно запустить OpenLane и прописать команду:  
**./flow.tcl -design <design\_name> -init\_design\_config -add\_to\_designs**  
Пример:

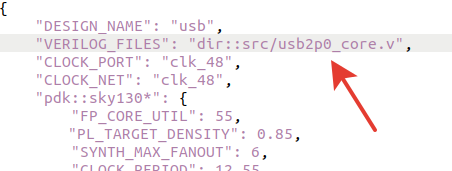


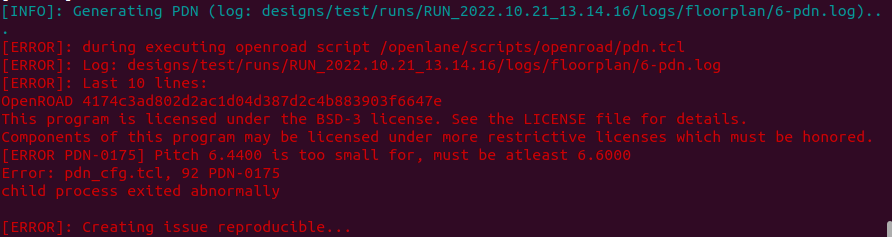
Мы создали дизайн с названием test, и создали для него базовый конфигурационный файл config.json, далее необходимо загрузить свой verilog файл в папку src.



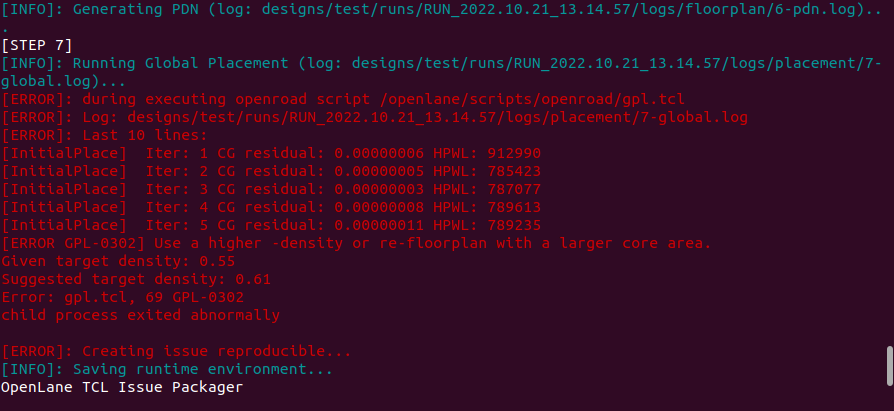
Какие сложности у меня возникли при добавлении своего дизайна:

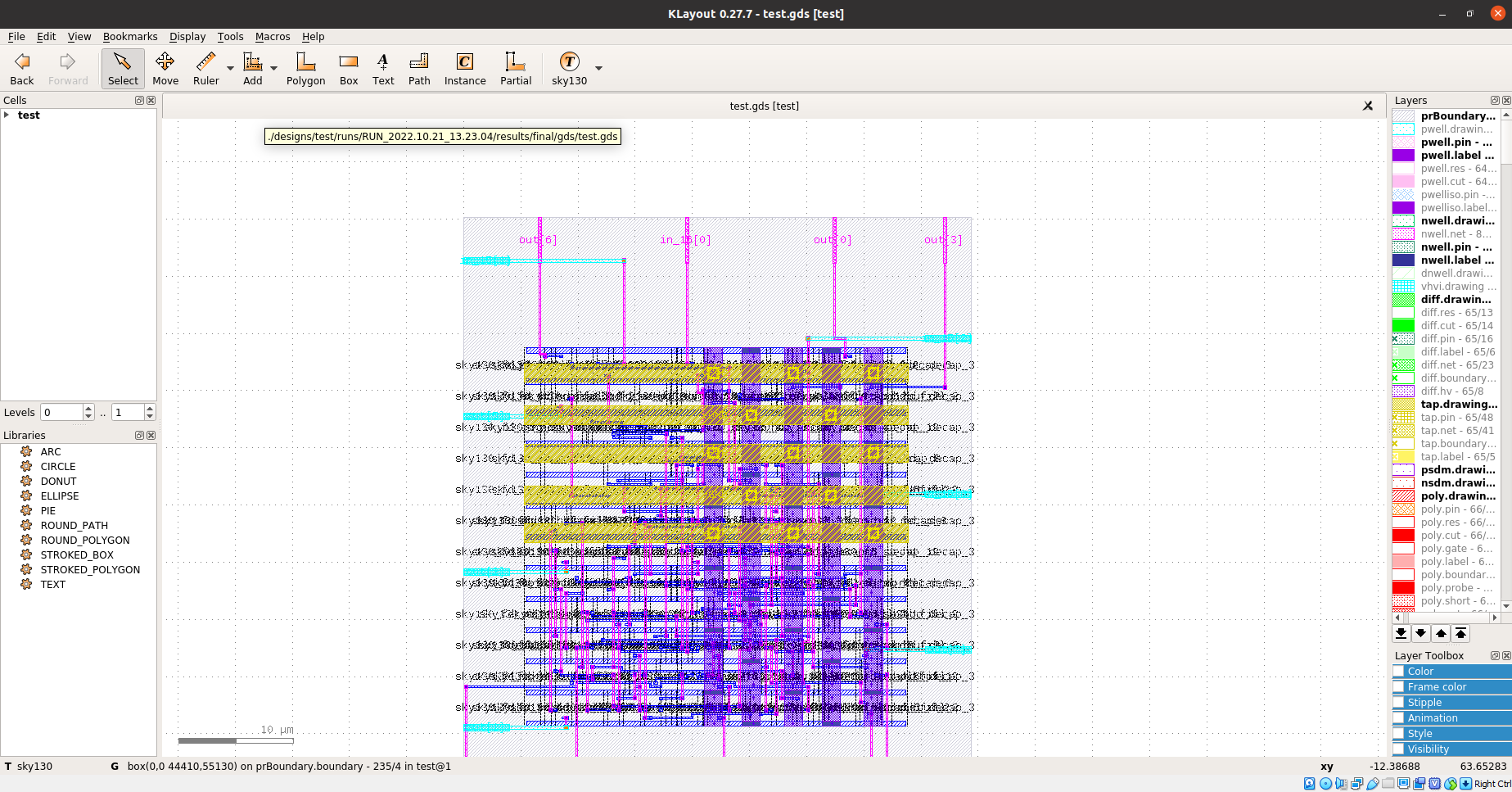


Не был найден модуль test. Необходимо либо подправить название модуля в верилоге, либо подправить config.json, и там прописать название верилог файла которое будет совпадать с модулем внутри. Пример как выглядит в конфиге usb:  
Далее появилась следующая проблема:



pitch 6.4400 был мал и просит выставить от 6.6. Для этого мы в конфигурационном файле дописываем пару параметров.  
“FP\_PDN\_VPITCH”: “7”,  
“FP\_PDN\_VPITCH”: “7”,  
  
Снова возникла проблема, на этот раз просит выставить другую плотность размещения ячеек:

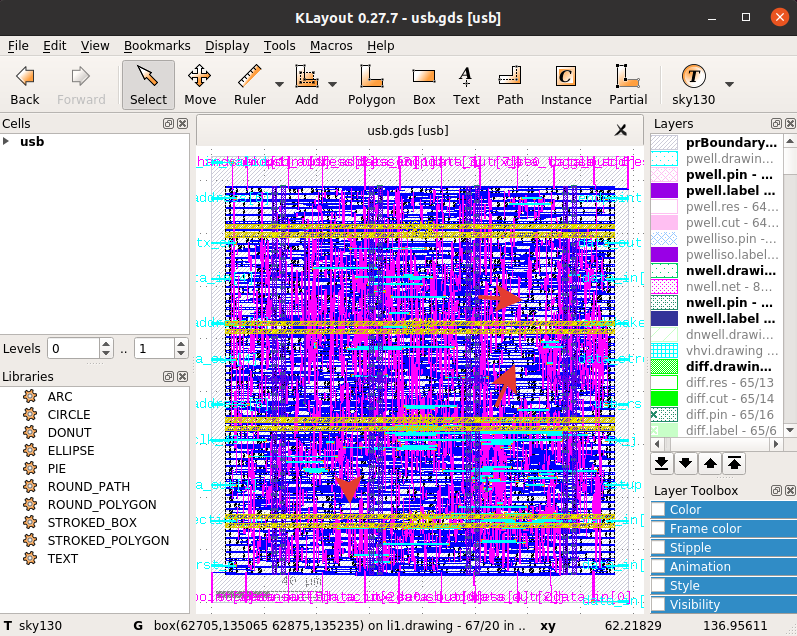


Выставил плотность размещения, но всё равно не удалось завершить размещения, для этого меняем “FP\_CORE\_UTIL”, т.к. по умолчанию он 50%, поставим 30%  
“PL\_TARGET\_DENSITY”: 0.61,  
“FP\_CORE\_UTIL”: 30,  
После этого поток успешно завершился, и я смог визуализировать gds файл:

**Практическая часть выполнения**

1) Визуализируем топологию usb которую мы получили запустив маршрут в самом начале, заменив <RUN\_\*\*\*> на свой  
  
**make mount**

**klayout -e -nn $PDK\_ROOT/sky130A/libs.tech/klayout/tech/sky130A.lyt \  
-l $PDK\_ROOT/sky130A/libs.tech/klayout/tech/sky130A.lyp \  
./designs/usb/runs/<RUN\_\*\*\*>/results/final/gds/usb.gds**



Визуально осматриваем и видим большое количество белого, Это не логические ячейки, там находятся decap ячейки и тд для заполнения пустот.

Decap - Ячейки представляют собой временные конденсаторы, добавленные в конструкцию между шинами питания и заземления для противодействия функциональным сбоям из-за динамического IR drop.

2) Попробуем сжать нашу схему usb, путём внесения изменений в конфигурационный файл схемы.  
  
Для этого будем использовать 2 параметра

“FP\_CORE\_UTIL”  
“PL\_TARGET\_DENSITY”  
Первый отвечает за процент использования площади ядра, второй за плотность размещения ячеек. Более подробно про них можете прочесть в документации:  
<https://openlane.readthedocs.io/en/latest/reference/configuration.html>



Ваша задача попробовать подобрать эти параметры чтобы схема меньше тратила свободного пространства и была плотнее. Визуализировать и посмотреть через программу klayout.

p.s. при большом значении “PL\_TARGET\_DENSITY” время трассировки существенно увеличивается, поэтому начинайте постепенно увеличивать.  
  
  
3) Загрузить свой дизайн в OpenLane