

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

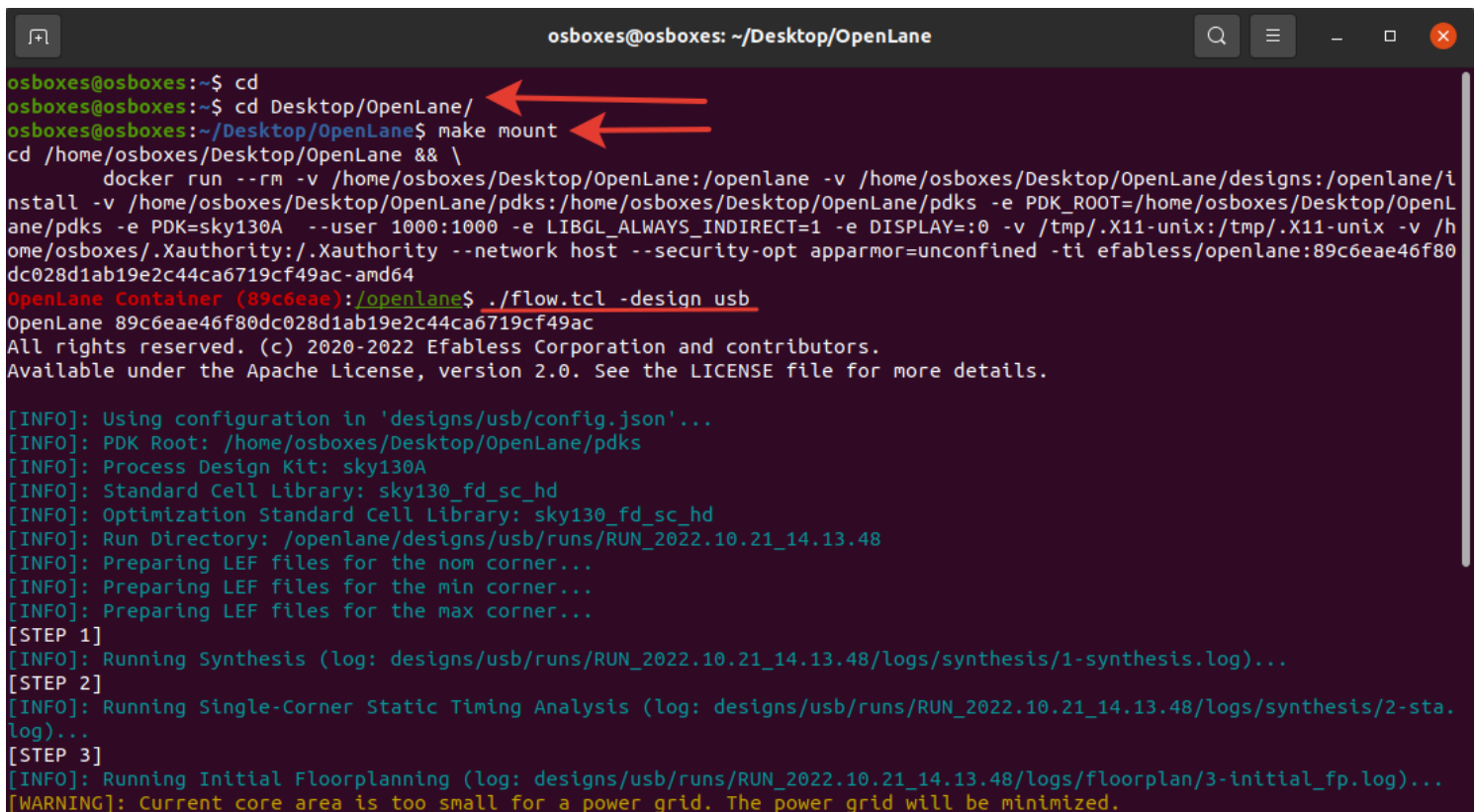
cd /Desktop/OpenLane/

3) Запустить OpenLane в докере с помощью

make mount

4)А теперь приступим к запуску flow.tcl


./flow.tcl -design usb



```
osboxes@osboxes: ~/Desktop/OpenLane
osboxes@osboxes:~$ cd
osboxes@osboxes:~$ cd Desktop/OpenLane/
osboxes@osboxes:~/Desktop/OpenLane$ make mount
cd /home/osboxes/Desktop/OpenLane && \
    docker run --rm -v /home/osboxes/Desktop/OpenLane:/openlane -v /home/osboxes/Desktop/OpenLane/designs:/openlane/i
ninstall -v /home/osboxes/Desktop/OpenLane/pdks:/home/osboxes/Desktop/OpenLane/pdks -e PDK_ROOT=/home/osboxes/Desktop/OpenL
ane/pdks -e PDK=sky130A --user 1000:1000 -e LIBGL_ALWAYS_INDIRECT=1 -e DISPLAY=:0 -v /tmp/.X11-unix:/tmp/.X11-unix -v /h
ome/osboxes/.Xauthority:/Xauthority --network host --security-opt apparmor=unconfined -ti efabless/openlane:89c6eae46f80
dc028d1ab19e2c44ca6719cf49ac-amd64
OpenLane Container (89c6eae):/openlane$ ./flow.tcl -design usb
OpenLane 89c6eae46f80dc028d1ab19e2c44ca6719cf49ac
All rights reserved. (c) 2020-2022 Efabless Corporation and contributors.
Available under the Apache License, version 2.0. See the LICENSE file for more details.

[INFO]: Using configuration in 'designs/usb/config.json'...
[INFO]: PDK Root: /home/osboxes/Desktop/OpenLane/pdks
[INFO]: Process Design Kit: sky130A
[INFO]: Standard Cell Library: sky130_fd_sc_hd
[INFO]: Optimization Standard Cell Library: sky130_fd_sc_hd
[INFO]: Run Directory: /openlane/designs/usb/runs/RUN_2022.10.21_14.13.48
[INFO]: Preparing LEF files for the nom corner...
[INFO]: Preparing LEF files for the min corner...
[INFO]: Preparing LEF files for the max corner...
[STEP 1]
[INFO]: Running Synthesis (log: designs/usb/runs/RUN_2022.10.21_14.13.48/logs/synthesis/1-synthesis.log)...
[STEP 2]
[INFO]: Running Single-Corner Static Timing Analysis (log: designs/usb/runs/RUN_2022.10.21_14.13.48/logs/synthesis/2-sta.
log)...
[STEP 3]
[INFO]: Running Initial Floorplanning (log: designs/usb/runs/RUN_2022.10.21_14.13.48/logs/floorplan/3-initial_fp.log)...
[WARNING]: Current core area is too small for a power grid. The power grid will be minimized.
```

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.

	kareefardi Fix <code>set_odb</code> invocation after adding obstructions (#1408)	✓ 89c6eae 2 days ago	🕒 1,592 commits
📁 .github	Startup Fixes (#1344)		25 days ago
📁 configuration	Elaboration, Signoff SDC Configuration Variables (#1406)		4 days ago
📁 dependencies	[BOT] Update PDK (#1389)		15 days ago
📁 designs	Elaboration, Signoff SDC Configuration Variables (#1406)		4 days ago
📁 docker	Elaboration, Signoff SDC Configuration Variables (#1406)		4 days ago
📁 docs	Elaboration, Signoff SDC Configuration Variables (#1406)		4 days ago
📁 regression_results	Documentation Restructure (#1337)		25 days ago
📁 scripts	Fix <code>set_odb</code> invocation after adding obstructions (#1408)		2 days ago
📁 tests	Use OpenDB as the Primary Layout Format (#1244)		18 days ago
📄 .flake8	Integrate Volare for PDK Building and Mangement, Part 2 (#1052)		6 months ago
📄 .gitattributes	checkers cleanup		2 years ago
📄 .gitignore	CI Fixes (#1022)		6 months ago
📄 .readthedocs.yml	Documentation Restructure (#1337)		25 days ago
📄 AUTHORS.md	Installation, Quickstart Guide and Sphynx Overhaul (#1259)		last month
📄 CONTRIBUTING.md	Documentation Restructure (#1337)		25 days ago
📄 Jenkinsfile	Fix tag names on Jenkins (#1119)		4 months ago
📄 LICENSE	Create LICENSE		2 years ago
📄 Makefile	Add Very Early support for GF180MCU + <code>open_pdk</code> Tweaks (#1251)		2 months ago
📄 README.md	Correct the installation link (#1384)		15 days ago
📄 env.py	Use OpenDB as the Primary Layout Format (#1244)		18 days ago
📄 flow.tcl	Support OpenROAD <code>set_dont_touch</code> (#1398)		5 days ago
📄 requirements.txt	Integrate Volare for PDK Building and Mangement, Part 2 (#1052)		6 months ago
📄 requirements_dev.txt	Support Multi-Arch Docker Builds (#1075)		5 months ago
📄 requirements_lint.txt	Python Cleanup (#864)		4 months ago
📄 run_designs.py	Python Cleanup (#864)		4 months ago
📄 run_issue_regressions.py	IR Drop Reports in the Signoff Stage (#1255)		2 months ago

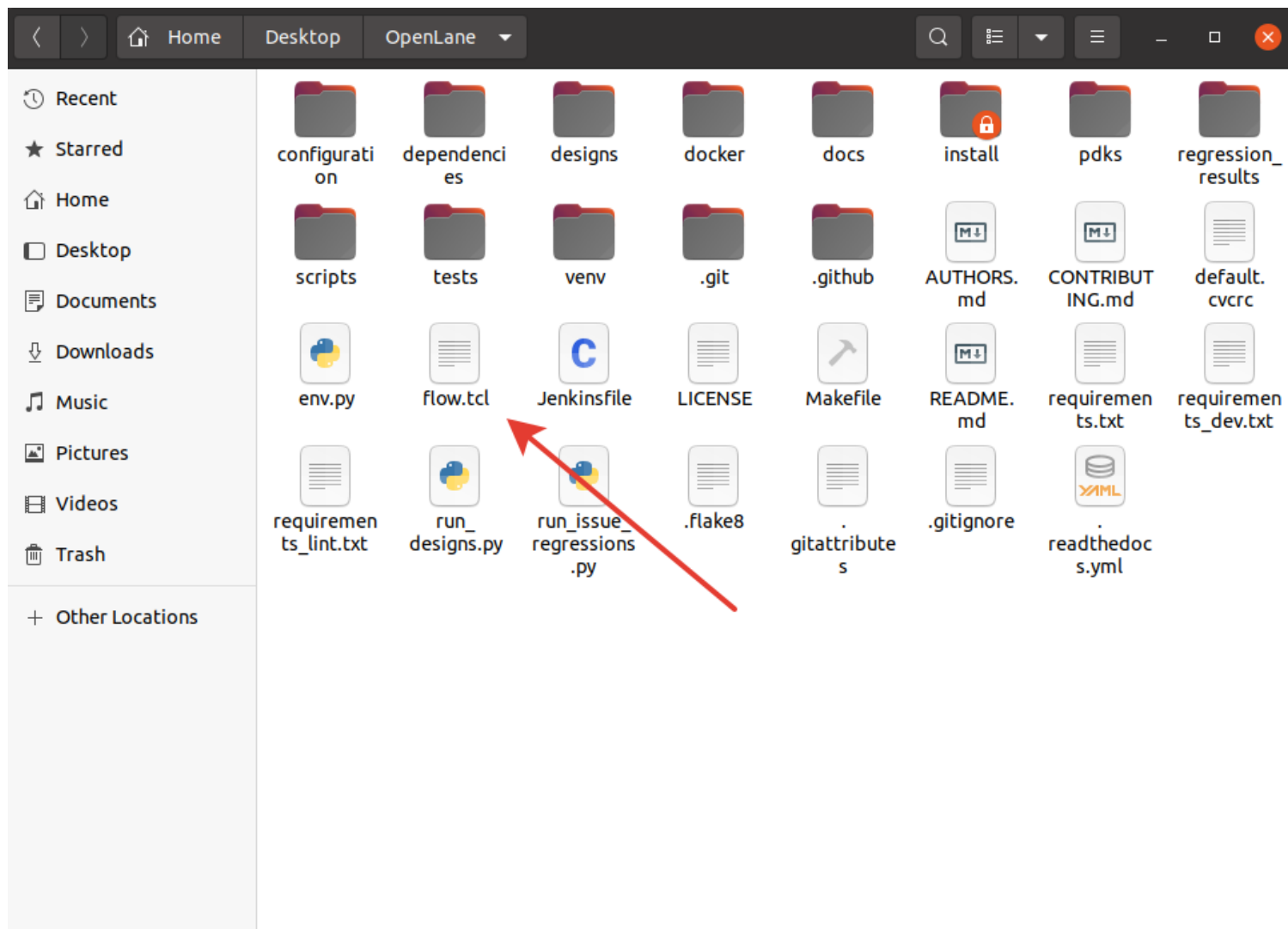
Для начала рассмотрим основные моменты которые нас интересуют из репозитория:

Папка scripts:

master OpenLane / scripts /	
kareefardi Fix set_odb invocation after adding obstructions (#1408)	
..	
config	Startup Fixes (#1344)
klayout	Documentation Restructure (#1337)
magic	Use OpenDB as the Primary Layout Format (#1244)
odbpy	Fix remove_buffers.py (#1396)
openroad	Support OpenROAD set_dont_touch (#1398)
report	Use OpenDB as the Primary Layout Format (#1244)
synth_exp	Fix Strategy Names (#1211)
td_commands	Fix set_odb invocation after adding obstructions (#1408)
topModuleGen	RTD Fixes (#1228)
utils	Generate Timing Models (#1395)
yosys	Elaboration, Signoff SDC Configuration Variables (#1406)
base.sdc	Documentation Fixes (#1333)
compare_regression_design.py	Multi-Process Corner RCX (#978)
compare_regression_reports.py	Tie Cells at FP Stage, Improved Cell Padding (#1226)
count_lvs.py	Add Python Linting (#876)
drc_rosetta.py	Fix .rdb DRC output translation (#1133)
extract_antenna_count.py	Update OpenROAD + Cleanup Fallout (#1169)
extract_antenna_violators.py	Fix Antenna Checkers, Magic Script Enhancements (#1154)
gen_report_routing.py	Add Python Linting (#876)

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

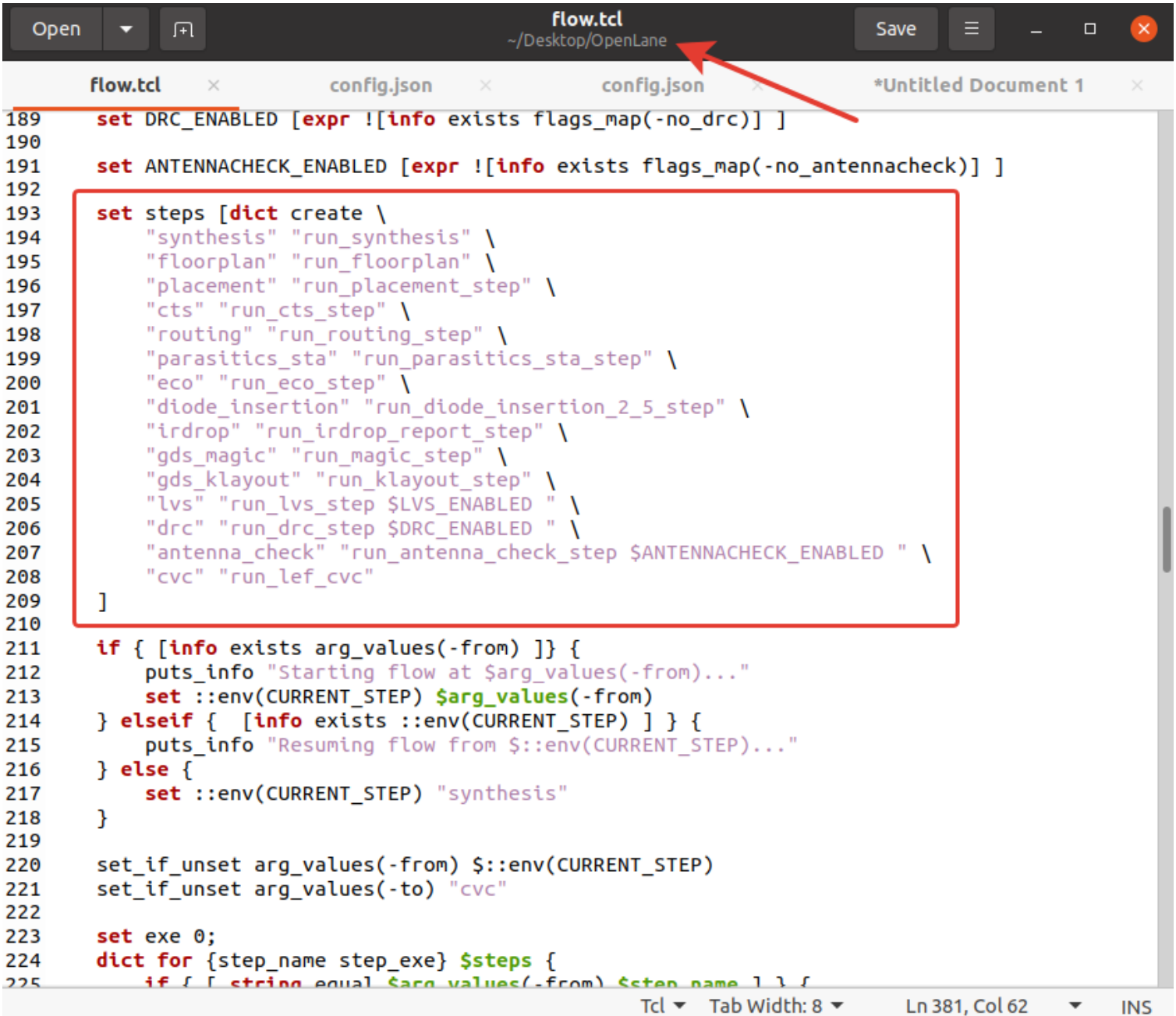
В OpenLane в главном разделе есть специальный скрипт на языке TCL “flow.tcl” в котором реализованно взаимодействие и вызов всех нужных для прохождения полного маршрута действий в правильном порядке. Нам нужно лишь запустить его указав название схемы, и весь маршрут будет проходить самостоятельно. Как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



```
189 set DRC_ENABLED [expr ![info exists flags_map(-no_drc)] ]
190
191 set ANTENNACHECK_ENABLED [expr ![info exists flags_map(-no_antennacheck)] ]
192
193 set steps [dict create \
194     "synthesis" "run_synthesis" \
195     "floorplan" "run_floorplan" \
196     "placement" "run_placement_step" \
197     "cts" "run_cts_step" \
198     "routing" "run_routing_step" \
199     "parasitics_sta" "run_parasitics_sta_step" \
200     "eco" "run_eco_step" \
201     "diode_insertion" "run_diode_insertion_2_5_step" \
202     "irdrop" "run_irdrop_report_step" \
203     "gds_magic" "run_magic_step" \
204     "gds_klayout" "run_klayout_step" \
205     "lvs" "run_lvs_step $LVS_ENABLED " \
206     "drc" "run_drc_step $DRC_ENABLED " \
207     "antenna_check" "run_antenna_check_step $ANTENNACHECK_ENABLED " \
208     "cvc" "run_lef_cvc"
209 ]
210
211 if { [info exists arg_values(-from)] } {
212     puts_info "Starting flow at $arg_values(-from)..."
213     set ::env(CURRENT_STEP) $arg_values(-from)
214 } elseif { [info exists ::env(CURRENT_STEP)] } {
215     puts_info "Resuming flow from $::env(CURRENT_STEP)..."
216 } else {
217     set ::env(CURRENT_STEP) "synthesis"
218 }
219
220 set_if_unset arg_values(-from) $::env(CURRENT_STEP)
221 set_if_unset arg_values(-to) "cvc"
222
223 set exe 0;
224 dict for {step_name step_exe} $steps {
225     if { [string equal $arg_values(-from) $step_name] } {
```

Tcl Tab Width: 8 Ln 381, Col 62 INS

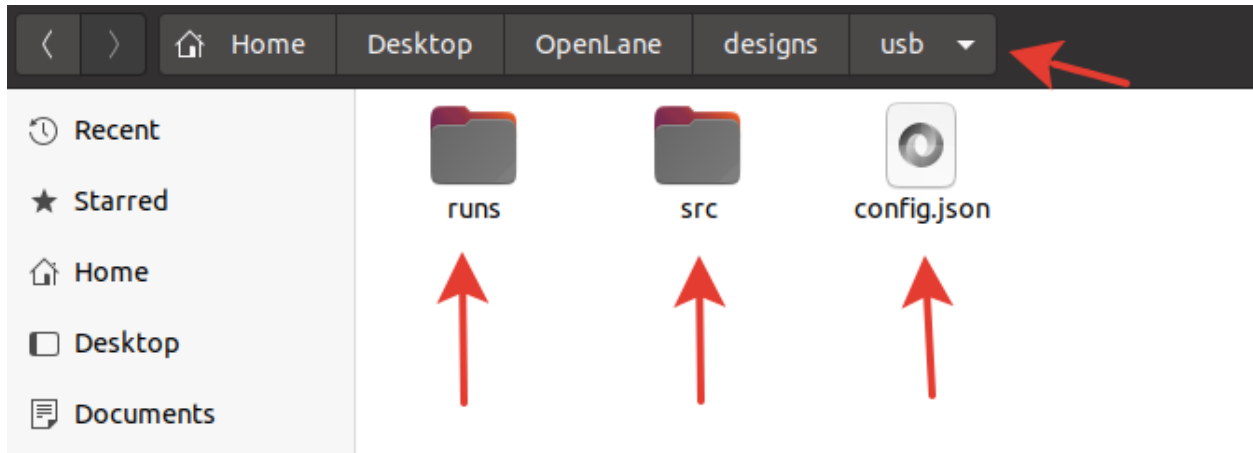
В дальнейшем эти “шаги” запускают скрипты из директории OpenLane/scripts и подтягивают различные параметры.

```
26 package require openlane; # provides the utils as well
27
28 proc run_placement_step {args} {
29     if { ! [ info exists ::env(PLACEMENT_CURRENT_DEF) ] } {
30         set ::env(PLACEMENT_CURRENT_DEF) $::env(CURRENT_DEF)
31     } else {
32         set ::env(CURRENT_DEF) $::env(PLACEMENT_CURRENT_DEF)
33     }
34
35     run_placement
36 }
37
38 proc run_cts_step {args} {
39     if { ! [ info exists ::env(CTS_CURRENT_DEF) ] } {
40         set ::env(CTS_CURRENT_DEF) $::env(CURRENT_DEF)
41     } else {
42         set ::env(CURRENT_DEF) $::env(CTS_CURRENT_DEF)
43     }
44
45     run_cts
46     run_resizer_timing
47     if { $::env(RSZ_USE_OLD_REMOVER) == 1 } {
48         remove_buffers_from_nets
49     }
50 }
51
52 proc run_routing_step {args} {
53     if { ! [ info exists ::env(ROUTING_CURRENT_DEF) ] } {
54         set ::env(ROUTING_CURRENT_DEF) $::env(CURRENT_DEF)
55     } else {
56         set ::env(CURRENT_DEF) $::env(ROUTING_CURRENT_DEF)
57     }
58     if { $::env(ECO_ENABLE) == 0 } {
59         run_routing
60     }
61 }
62
```

Tcl Tab Width: 8 Ln 381, Col 62 INS

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

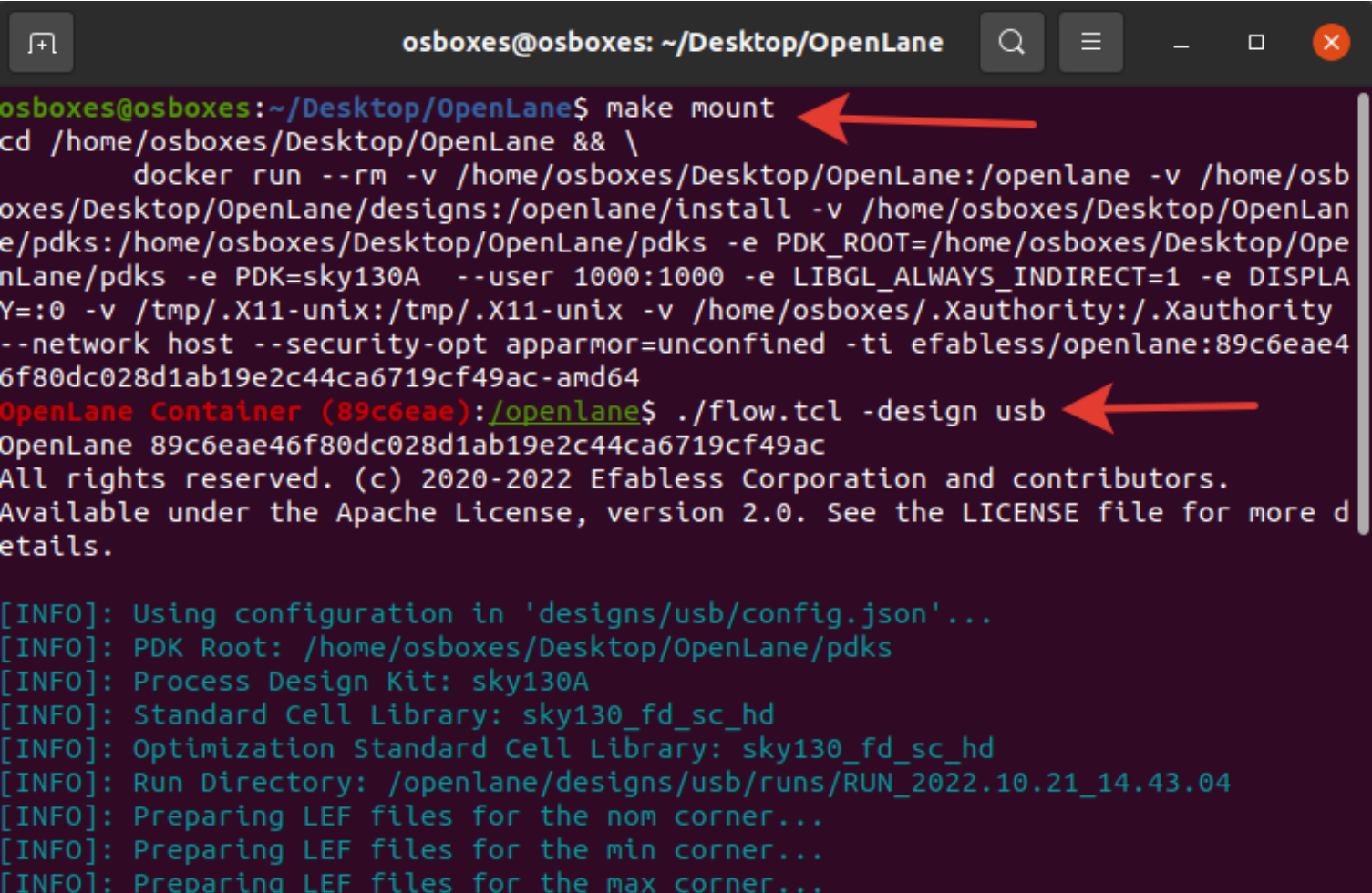
OpenLane запускается с помощью Докера.

Для начала необходимо перейти в директорию OpenLane:

cd /Desktop/OpenLane

Теперь нужно запустить OpenLane:

make mount



```
osboxes@osboxes: ~/Desktop/OpenLane
osboxes@osboxes:~/Desktop/OpenLane$ make mount
cd /home/osboxes/Desktop/OpenLane && \
    docker run --rm -v /home/osboxes/Desktop/OpenLane:/openlane -v /home/osboxes/Desktop/OpenLane/designs:/openlane/install -v /home/osboxes/Desktop/OpenLane/pdks:/home/osboxes/Desktop/OpenLane/pdks -e PDK_ROOT=/home/osboxes/Desktop/OpenLane/pdks -e PDK=sky130A --user 1000:1000 -e LIBGL_ALWAYS_INDIRECT=1 -e DISPLAY=:0 -v /tmp/.X11-unix:/tmp/.X11-unix -v /home/osboxes/.Xauthority:/home/osboxes/.Xauthority --network host --security-opt apparmor=unconfined -ti efabless/openlane:89c6eae46f80dc028d1ab19e2c44ca6719cf49ac-amd64
OpenLane Container (89c6eae):/openlane$ ./flow.tcl -design usb
OpenLane 89c6eae46f80dc028d1ab19e2c44ca6719cf49ac
All rights reserved. (c) 2020-2022 Efabless Corporation and contributors.
Available under the Apache License, version 2.0. See the LICENSE file for more details.

[INFO]: Using configuration in 'designs/usb/config.json'...
[INFO]: PDK Root: /home/osboxes/Desktop/OpenLane/pdks
[INFO]: Process Design Kit: sky130A
[INFO]: Standard Cell Library: sky130_fd_sc_hd
[INFO]: Optimization Standard Cell Library: sky130_fd_sc_hd
[INFO]: Run Directory: /openlane/designs/usb/runs/RUN_2022.10.21_14.43.04
[INFO]: Preparing LEF files for the nom corner...
[INFO]: Preparing LEF files for the min corner...
[INFO]: Preparing LEF files for the max corner...
```

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
```

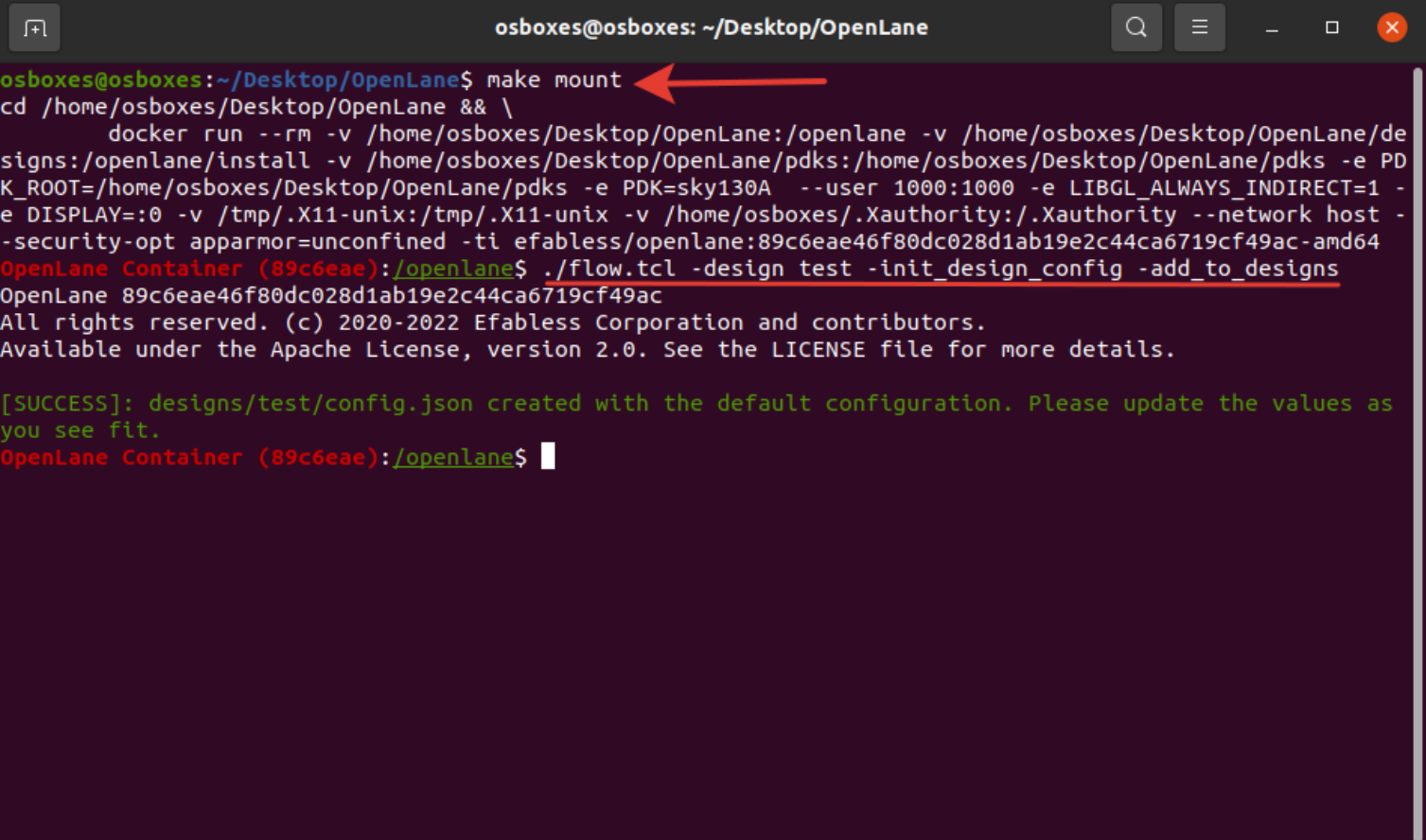
Данные параметры конфигурационного файла добавляются к основному который лежит в папке с дизайном

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`

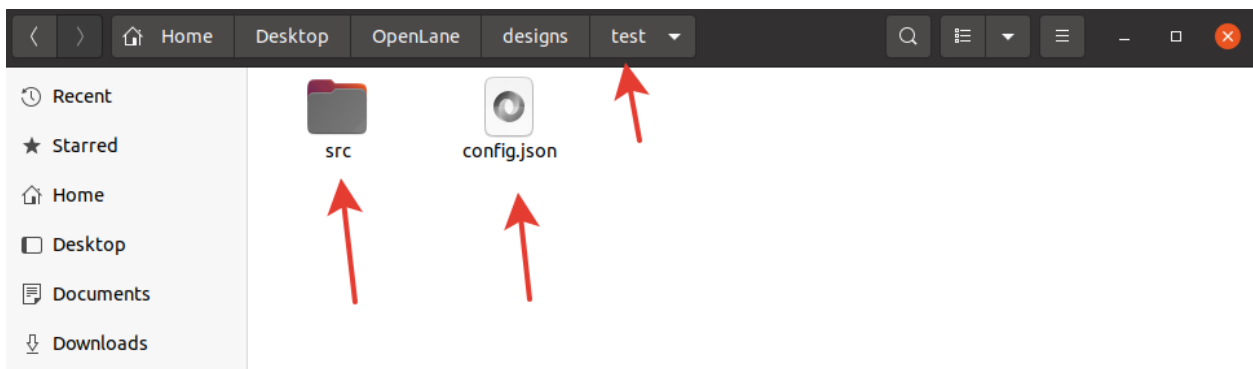
Пример:



```
osboxes@osboxes: ~/Desktop/OpenLane
osboxes@osboxes:~/Desktop/OpenLane$ make mount
cd /home/osboxes/Desktop/OpenLane && \
    docker run --rm -v /home/osboxes/Desktop/OpenLane:/openlane -v /home/osboxes/Desktop/OpenLane/designs:/openlane/install -v /home/osboxes/Desktop/OpenLane/pdks:/home/osboxes/Desktop/OpenLane/pdks -e PDK_ROOT=/home/osboxes/Desktop/OpenLane/pdks -e PDK=sky130A --user 1000:1000 -e LIBGL_ALWAYS_INDIRECT=1 -e DISPLAY=:0 -v /tmp/.X11-unix:/tmp/.X11-unix -v /home/osboxes/.Xauthority:/home/osboxes/.Xauthority --network host --security-opt apparmor=unconfined -ti efabless/openlane:89c6eae46f80dc028d1ab19e2c44ca6719cf49ac-amd64
OpenLane Container (89c6eae):/openlane$ ./flow.tcl -design test -init_design_config -add_to_designs
OpenLane 89c6eae46f80dc028d1ab19e2c44ca6719cf49ac
All rights reserved. (c) 2020-2022 Efabless Corporation and contributors.
Available under the Apache License, version 2.0. See the LICENSE file for more details.

[SUCCESS]: designs/test/config.json created with the default configuration. Please update the values as you see fit.
OpenLane Container (89c6eae):/openlane$
```

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:

```
osboxes@osboxes: ~/Desktop/OpenLane
[INFO]: Preparing LEF files for the nom corner...
[INFO]: Preparing LEF files for the min corner...
[INFO]: Preparing LEF files for the max corner...
[STEP 1]
[INFO]: Running Synthesis (log: designs/test/runs/RUN_2022.10.21_12.55.00/logs/synthesis/1-synthesis.log)...
[ERROR]: during executing: "yosys -c /openlane/scripts/yosys/synth.tcl -l /openlane/designs/test/runs/RUN_2022.10.21_12.55.00/logs/synthesis/1-synthesis.log |& tee /dev/null"
[ERROR]: Exit code: 1
[ERROR]: Last 10 lines:

[TCL: yosys -import] Command name collision: found pre-existing command `cd' -> skip.
[TCL: yosys -import] Command name collision: found pre-existing command `eval' -> skip.
[TCL: yosys -import] Command name collision: found pre-existing command `exec' -> skip.
[TCL: yosys -import] Command name collision: found pre-existing command `read' -> skip.
[TCL: yosys -import] Command name collision: found pre-existing command `trace' -> skip.

1. Executing Verilog-2005 frontend: /openlane/designs/test/src/test.v
ERROR: No such module: test
child process exited abnormally

[INFO]: Saving current set of views in 'designs/test/runs/RUN_2022.10.21_12.55.00/results/final'..
[INFO]: Generating final set of reports...
[INFO]: Created manufacturability report at 'designs/test/runs/RUN_2022.10.21_12.55.00/reports/manufacturability.rpt'.
[INFO]: Created metrics report at 'designs/test/runs/RUN_2022.10.21_12.55.00/reports/metrics.csv'.
```

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:

```
{
  "DESIGN_NAME": "usb",
  "VERILOG_FILES": "dir::src/usb2p0_core.v",
  "CLOCK_PORT": "clk_48",
  "CLOCK_NET": "clk_48",
  "pdk::sky130*": {
    "FP_CORE_UTIL": 55,
    "PL_TARGET_DENSITY": 0.85,
    "SYNTH_MAX_FANOUT": 6,
    "CLOCK_PERIOD": 12.55
  }
}
```

Далее появилась следующая проблема:

```
[INFO]: Generating PDN (log: designs/test/runs/RUN_2022.10.21_13.14.16/logs/floorplan/6-pdn.log)..
[ERROR]: during executing openroad script /openlane/scripts/openroad/pdn.tcl
[ERROR]: Log: designs/test/runs/RUN_2022.10.21_13.14.16/logs/floorplan/6-pdn.log
[ERROR]: Last 10 lines:
OpenROAD 4174c3ad802d2ac1d04d387d2c4b883903f6647e
This program is licensed under the BSD-3 license. See the LICENSE file for details.
Components of this program may be licensed under more restrictive licenses which must be honored.
[ERROR PDN-0175] Pitch 6.4400 is too small for, must be atleast 6.6000
Error: pdn_cfg.tcl, 92 PDN-0175
child process exited abnormally

[ERROR]: Creating issue reproducible...
```

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",

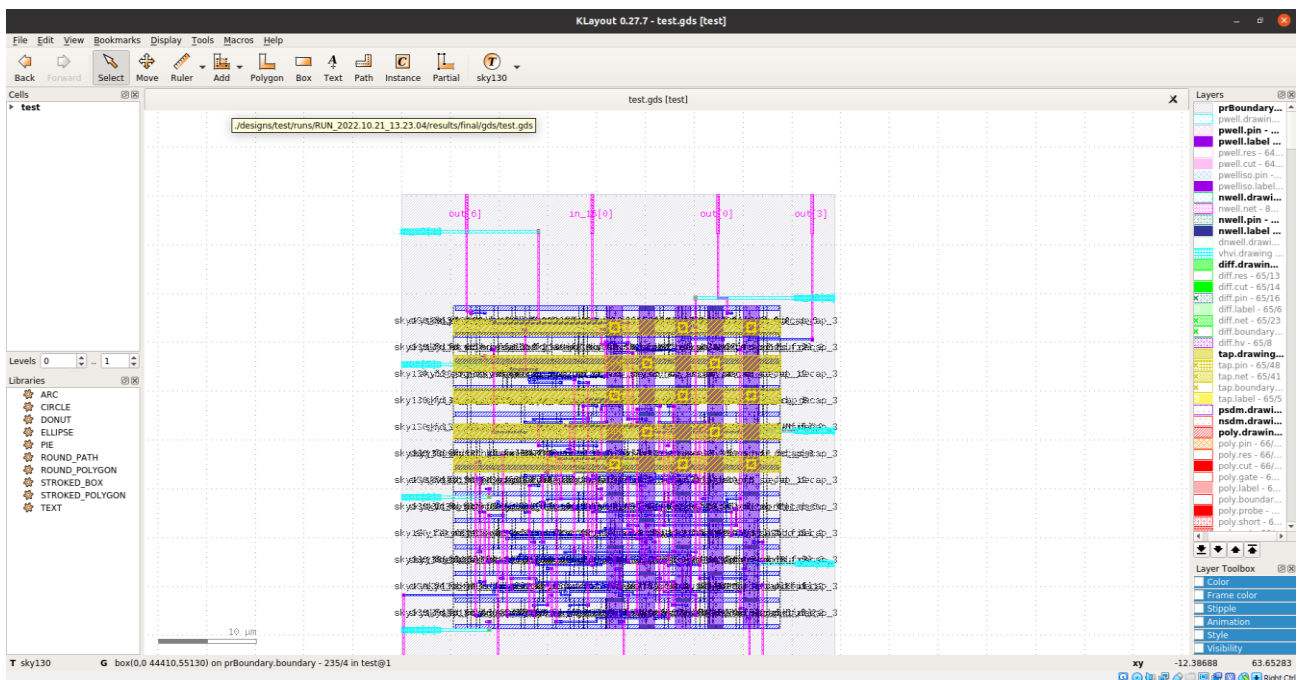
Снова возникла проблема, на этот раз просит выставить другую плотность размещения ячеек:

```
[INFO]: Generating PDN (log: designs/test/runs/RUN_2022.10.21_13.14.57/logs/floorplan/6-pdn.log)..  
[STEP 7]  
[INFO]: Running Global Placement (log: designs/test/runs/RUN_2022.10.21_13.14.57/logs/placement/7-global.log)..  
[ERROR]: during executing openroad script /openlane/scripts/openroad/gpl.tcl  
[ERROR]: Log: designs/test/runs/RUN_2022.10.21_13.14.57/logs/placement/7-global.log  
[ERROR]: Last 10 lines:  
[InitialPlace] Iter: 1 CG residual: 0.00000006 HPWL: 912990  
[InitialPlace] Iter: 2 CG residual: 0.00000005 HPWL: 785423  
[InitialPlace] Iter: 3 CG residual: 0.00000003 HPWL: 787077  
[InitialPlace] Iter: 4 CG residual: 0.00000008 HPWL: 789613  
[InitialPlace] Iter: 5 CG residual: 0.00000011 HPWL: 789235  
[ERROR GPL-0302] Use a higher -density or re-floorplan with a larger core area.  
Given target density: 0.55  
Suggested target density: 0.61  
Error: gpl.tcl, 69 GPL-0302  
child process exited abnormally  
  
[ERROR]: Creating issue reproducible...  
[INFO]: Saving runtime environment...  
OpenLane TCL Issue Packager
```

Выставил плотность размещения, но всё равно не удалось завершить размещения, для этого меняю "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:

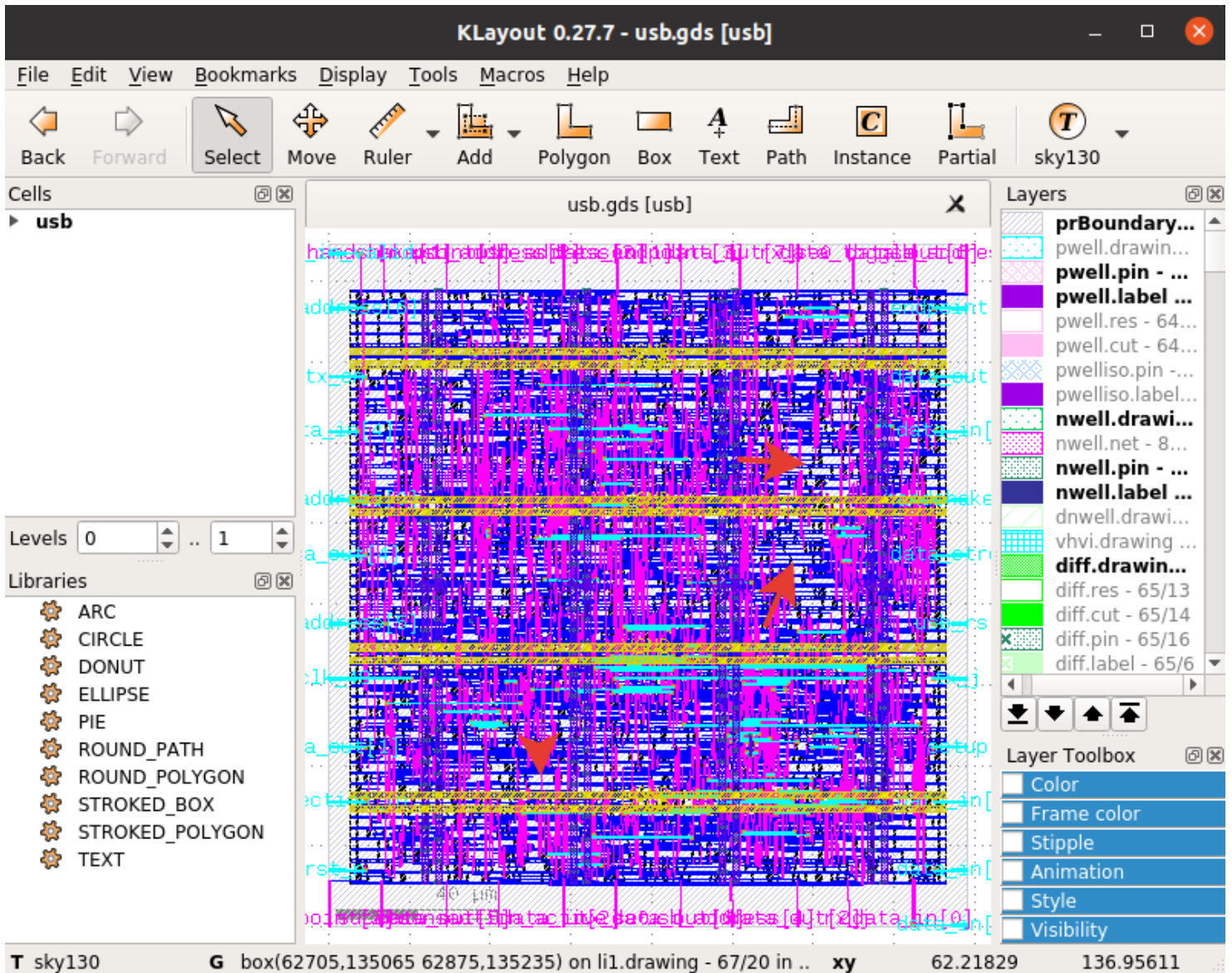


Практическая часть выполнения

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 \
-I $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 десар ячейки и тд для заполнения пустот.

Десар - 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>



```
1 {
2   "DESIGN_NAME": "usb",
3   "VERILOG_FILES": "dir::src/usb2p0_core.v",
4   "CLOCK_PORT": "clk_48",
5   "CLOCK_NET": "clk_48",
6   "pdk::sky130*": {
7     "FP_CORE_UTIL": 2,
8     "PL_TARGET_DENSITY": 2,
9     "SYNTH_MAX_FANOUT": 6,
10    "CLOCK_PERIOD": 12.55,
11    "scl::sky130_fd_sc_hd": {
12      "CLOCK_PERIOD": 15,
13      "SYNTH_STRATEGY": "DELAY 0"
14    }
15  }
16 }
```

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