Chapter 8 - Tapeout

Course authors (Git file)

1. Open-source Tapeout informations from IHP
2. READMEs
3. Read-the-docs
4. The last steps to a successful Tapeout
5. Contribute to open-source

Section 1

Open-source Tapeout informations from IHP

Open-source Tapeout informations from IHP

This chapter provides information about the Tapeout at IHP. Please follow the links to tag along with the chapter.

IHP - Repositories structure

- The designs for the open-source Tapeouts are collected as the “Open DesignLib”.
- During the process of each Tapeout there are dedicated repositories by the names TO_monthyear.
- After Tapeout these designs shall be merged into the Open DesignLib repository.

PDK and Open DesignLib

- Open.source PDK:

https://github.com/IHP-GmbH/IHP-Open-PDK

- Open DesignLib:

https://github.com/IHP-GmbH/IHP-Open-DesignLib

TapeOut repositories (TO)

- Tapeout May 2024:

https://github.com/IHP-GmbH/TO_May2024

- Tapeout Nov 2024:

https://github.com/IHP-GmbH/TO_Nov2024

- Tapeout Dec 2024:

https://github.com/IHP-GmbH/TO_Dec2024

Section 2

READMEs

README file in TO repositories

Important notes:

- Always look for the most actual and stable information
- Tools and workflows change faast
- Design rules don’t change fast (if ever)
- Most parts of the PDK are long term stable

Submission process

Screenshot from the README TO_DEC2024:

Submission process

To submit for our OpenMPW run you have to have a valid github account. Make a fork of this repository and then create a separate directory for your design next to the `ExampleDesign` (you can also make a copy and rename it). Structure your data according to our recommendations, update the documentation and push your files to your fork, then make a pull request.

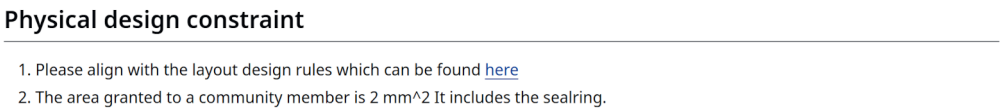
> Caution
>
> On each PR a github action will be triggered to run a minimal DRC precheck (rejection test). Please consider it and do not upload many `gds` files.

Once you make a PR a github action will run a minimum set of DRC checks on each `gds` and `gds.zip` file. If the test passes it means that your design is manufacturable at our pilot line not ensuring the reliability. An example of a failure is shown on the following figure

Figure 1: Submission process

Physical constraints

Screenshot from the README TO_DEC2024:



Physical design constraint

1. Please align with the layout design rules which can be found here
2. The area granted to a community member is 2 mm^2 It includes the sealring.
3. The sealring can be found among KLayout PyCells.

Figure 2: Physical design constraints

Project structure

Screenshot from the README TO_DEC2024:

Directory structure

If you are a designer, we propose the following directory structure, which we and the community would appreciate you using. Please ensure that the design you submit is reproducible, meaning it should include all the information necessary to replicate the design.

```
<design_name>
├ design_data
│ └ tool1/format1/step1
│   └ data
│ └ tool2/format2/setp2
│   └ data
├ doc
│ ├ specification
│ ├ ...
└ val <- validation/verification >
```

The first segmentation separates the `design data` from a `documentation` and `verification/validation data`.

Figure 3: Directories in a project

Design data structure

Screenshot from the README TO_DEC2024:

Design data directory structure

The `design data` should be structured using tool/format/step specific scheme.

Here you can find some examples:

Example1

```
design_data
└ xschem
└ ngspice
└ klayout
└ drc
└ lvs
└ gds
```

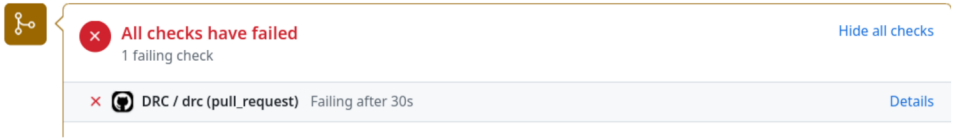
Example2

```
design_data
└ verilog
└ sdc
└ lef
└ gds
└ lib
└ drc
└ lvs
└ reports
└ log
```

DRC

During a submission to a Tapeout the design gets checked with a DRC via Github Actions.

Once you make a PR a github action will run a minimum set of DRC checks on each `gds` and `gds.zip` file. If the test passes it means that your design is manufacturable at our pilot line not ensuring the reliability. An example of a failure is shown on the following figure



The detailed report can be downloaded from a link, which can be found at the end of the section `Details->Archive DRC Results` as show on the image:

Figure 5: DRC via Github actions

Section 3

Read-the-docs

Read-the-docs

- Open PDK Docu:

https://ihp-open-pdk-docs.readthedocs.io/en/latest/index.html

- Open DesignLib Docu:

https://ihp-open-ip.readthedocs.io/en/latest/

README versus Read-the-docs

- The README version is a extract of the full documentation (read-the-docs) for the DesignLib.
- The read-the-docs version contains the Tapeout calender and further info

Tapeout calendar

Welcome to IHP-Open-DesignLib documentation!

IHP-Open-DesignLib is repository, which contains open source IC designs using IHP SG13G2 BiCMOS processs. It is also a central point for design fabrication under the concept of IHP Free MPW runs funded by a public German project FMD-QNC (16ME083). Project funds can be used exclusively to produce chip designs for non-commercial activities, such as university education, research projects, and others. In the project, a continuation for the provision of free area for the open source community is to be worked out.

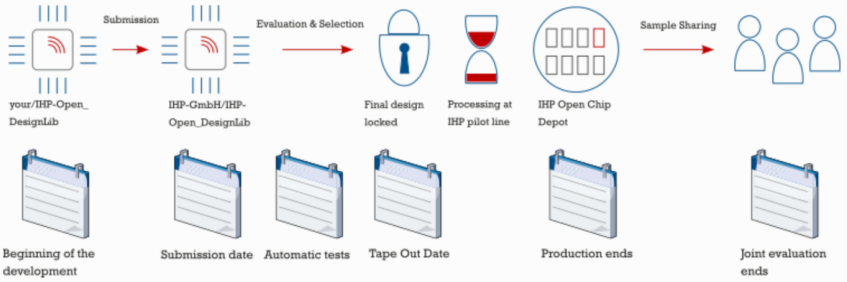
Tape In date	10 May 2024	11 Nov 2024	22 Nov 2024	07 Apr 2025	09 May
Technology	SG13G2	SG13CMOS	SG13G2	SG13G2	SG13G2
Area available [mm^2]	10	220	20	140	30

11 Nov 2024	22 Nov 2024	07 Apr 2025	09 May 2025	18 Jul 2025	15 Sep 2025
SG13CMOS	SG13G2	SG13G2	SG13G2	SG13G2	SG13CMOS
220	20	140	30	30	220

Figure 6: Open DesignLib Tapeout Calendar

More submission information

The overview of the submission process is shown on the following figure.



The submission process contains a few steps, where some of them are mandatory and crucial:

1. Project development phase. At the beginning specifications an criteria will be defined by PDK status, later specifications from sponsors might be possible

Figure 7: Submission process detailed

Section 4

The last steps to a successful Tapeout

The last steps to a successful Tapeout

A few words about the finish of a Tapeout-ready GDS:

- A successfull Tapeout for a producable, correct and working micorchip contains more steps and knowdledge. It is out of scope of the course to explain and tutor these all details about this knowledge.
- Most of these steps will integrate into ORFS soon in a easy-to-use way eventually.

Topic to watch in design

Till then, here is a (not complete) list of important topics:

- Placement and instanciation of the Input-Outputs.
- Metall fills
- Sealring construction
- Automated DRC free

Github Actions

- Tapeout Nov 2024 Actions drc.yml:

https://github.com/IHP-GmbH/TO_Nov2024/blob/main/.github/workflows/drc.yml

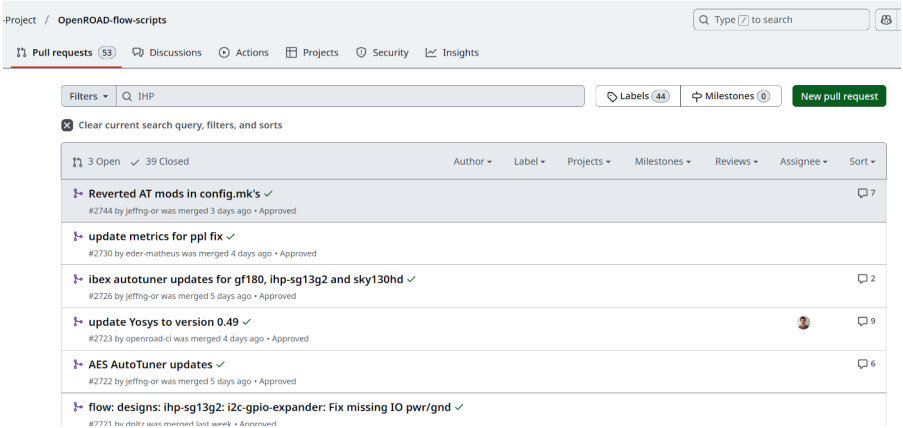
- Tapeout Nov 2024 Actions runs:

https://github.com/IHP-GmbH/TO_Nov2024/actions

Active development in ORFS

Pull requests in ORFS with the search IHP in it:

https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts/pulls?q=IHP+



Issues in ORFS (open and IHP in it):

https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts/issues?q=is%3Aissue%20state%3Aopen%20IHP

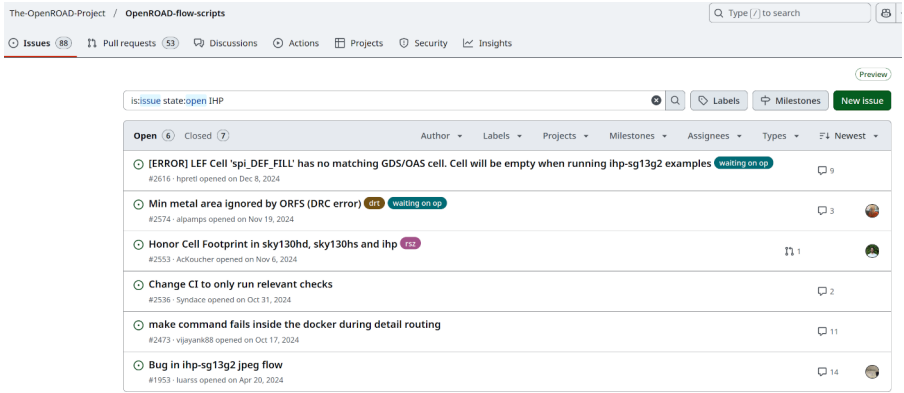
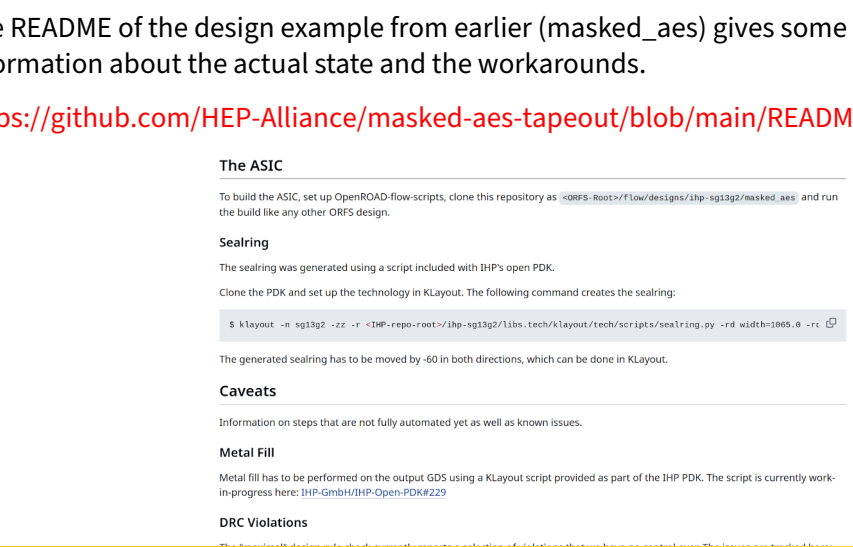


Figure 9: ORFS issues, open and IHP in it

Workarounds (example masked_aes)

The README of the design example from earlier (masked_aes) gives some basic, but sufficient information about the actual state and the workarounds.

https://github.com/HEP-Alliance/masked-aes-tapeout/blob/main/README.md



The ASIC

To build the ASIC, set up OpenROAD-flow-scripts, clone this repository as `<ORFS-Root>/flow/designs/ihp-sg13g2/masked_aes` and run the build like any other ORFS design.

Sealring

The sealring was generated using a script included with IHP's open PDK.

Clone the PDK and set up the technology in KLayout. The following command creates the sealring:

```
$ klayout -n sg13g2 -zz -r <IHP-repo-root>/ihp-sg13g2/libs.tech/klayout/tech/scripts/sealring.py -rd width=1065.0 -rd
```

The generated sealring has to be moved by -60 in both directions, which can be done in KLayout.

Caveats

Information on steps that are not fully automated yet as well as known issues.

Metal Fill

Metal fill has to be performed on the output GDS using a KLayout script provided as part of the IHP PDK. The script is currently work-in-progress here: IHP-GmbH/IHP-Open-PDK#229

DRC Violations

Before your Tapeout

If you plan to do a Tapeout with open-source to a Shuttlerun at IHP:

- Get in touch with the people at IHP before submitting a GDS.
- Inform yourself about the actual changes in the tools, the PDK and the submission repositories.
- Look for recent examples from the CoreExpert Group.

Non public and non open-source designs

IMPORTANT NOTICE:

- Using open-source doesn't mean you have to publish your work in open-source!

With using the open-source EDA tools and the open-source IHP PDK it is possible

- to design and fabricate closed source microchips at IHP.
- to not make a design public.

If you plan to do this: Talk to IHP!

Section 5

Contribute to open-source

Contributing with Git

If you waant to start contributing to open-source (maybe to this course?):

- Create a Git account
- Join a discussion about an issue.
- When you discover an error, writ4e an issue yourself.

It is really not much more then a ticket-system vi email.

The Feedback to this course

Your feedback will become issues in the course repository. Join in the discussion if you want to conrtribute:

https://github.com/OS-EDA/Course/issues?q=is%3Aissue

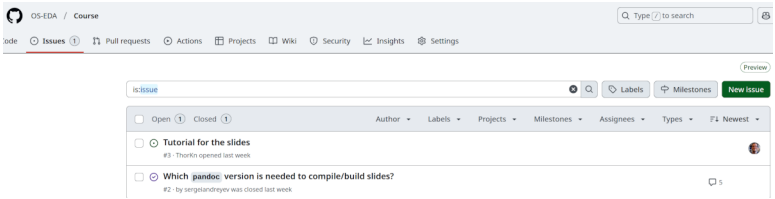


Figure 11: Issue tracking in the course repository