



Leibniz Institute
for high
performance
microelectronics

Welcome to the IHP Analog Certificate Course (Trial Run) 2025

Phillip Ferreira Baade–Pedersen (Design Engineer)

Dr. Ing. Christian Wittke (Scientist)

Frankfurt (Oder), IHP - 19/05/2025

Projects: BMBF → FMD-QNC (16ME0831)

Agenda For today



Leibniz Institute
for high
performance
microelectronics

09:00 - 09:30 | Welcome

Welcome Committee / Getting Ready

09:30 - 10:00 | Course Overview & Weekly Plan

Introduction to the course and instructors

Expectations and goals

10:00 - 11:00 | IHP & The Open PDK

11:00 - 12:00 | Laying The Groundwork

Creating A simple testbench in Xschem

First hands on exercises

12:00 - 13:00 | Lunch Brake | Catching Up

13:00 - 15:00 | Basics & Fundamental In Klayout

*Playing around and getting comfortable
with KLayout*

15:00 – 15:30 | Coffee Break | Caching Up

15:30 - 17:00 | Hands-On Session

Expert Talks: IHP Open PDK Group & Guests

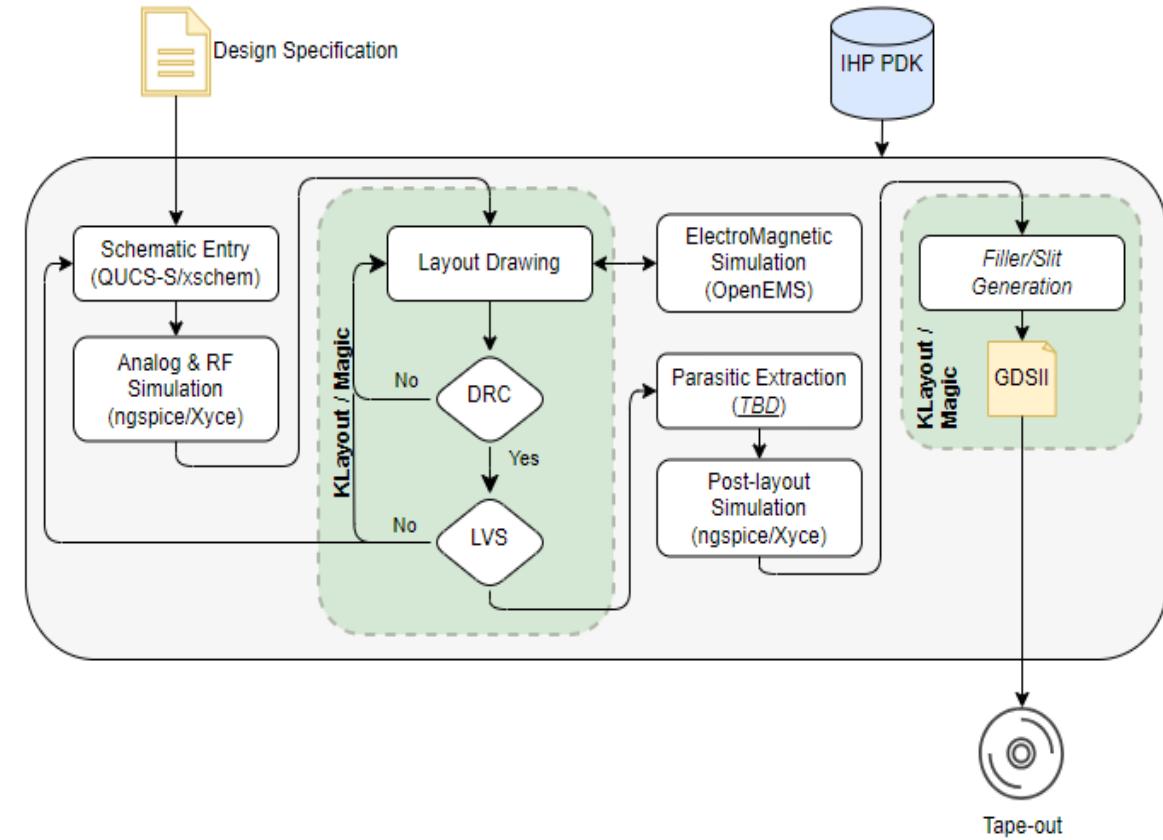


Prabhat Kumar Dubey (Scientist): Device Modelling and Model generation inside the Open PDK

Dr. Volker Muehlhaus: Electromagnetic Simulations in OS with OpenEMS and IHP stackup

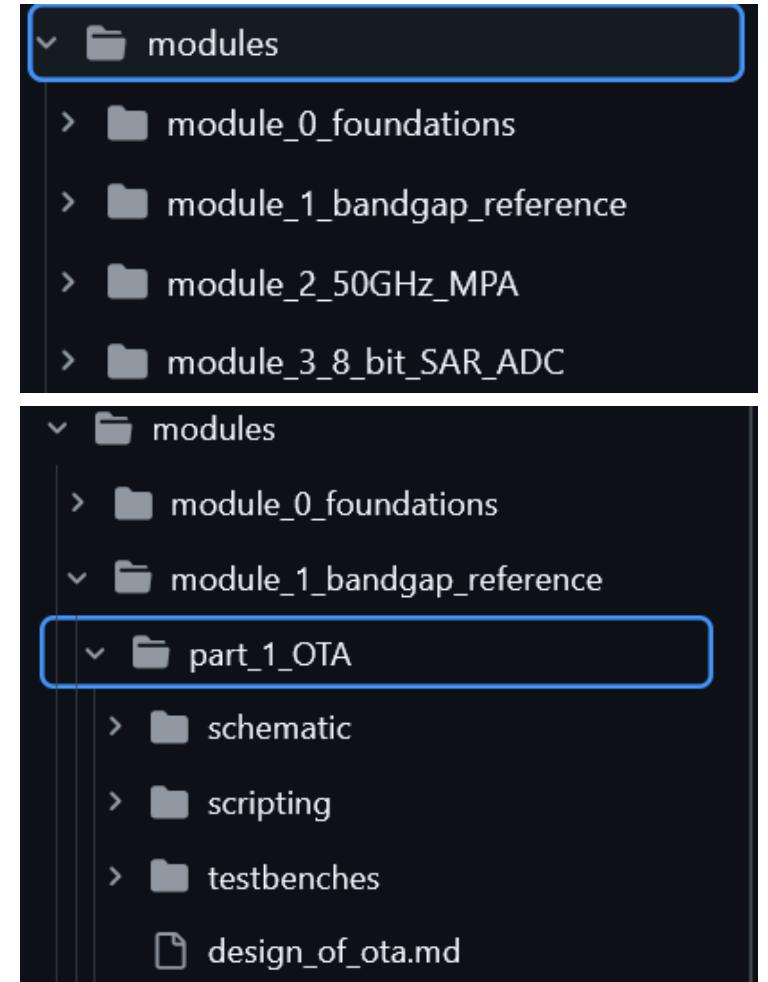
About The Scope Of This Course

- 0 Navigating in the analog OS echo system
- 0 Going from design to layout
- 0 Setting up a professional workflow similar to conventional chip design
- 0 Understand the gaps yet to be closed
- ~~-0 Design Circuits~~



The Structure of the Course:

- 0 Module based course
- 0 All content is available in the repository on your machines
- 0 Use the markdown files as cheat sheets or if you get stuck
- 0 Most of the designs should run with no additional setup
- 0 Its recommended to copy the structure seen in the repo
(alternatively create duplicated folders)





Leibniz Institute
for high
performance
microelectronics

Live Demonstration Of The Repository

Now, Let's Get to Know Each Other!



- 0 Have you worked with any of the tools we'll be using?
- 0 What challenges have you faced in similar projects?
- 0 What's your background and experience in this field?
- 0 What are you hoping to learn from this course?

IHP in a nutshell



- 0 IHP is the European research and innovation centre for silicon-based systems, ultrahigh-frequency circuits and technologies,
- 0 Unique selling point of a 200mm pilot line for state-of-the-art BiCMOS technologies, operated under industry-like conditions, 24/7, for the provision of prototypes and low-volume production runs.
- 0 Qualified technological platform with direct access for science and industry
- 0 Vertical structure from material research to system architecture
- 0 350+ employees, 40+ nationalities



Vision

"We create foundations and prototype applications based on future silicon-based technologies and systems for a digitalized and networked world as well as for the sustainable preservation of our natural living conditions."

130nm SiGe BiCMOS Technologies for RF Applications

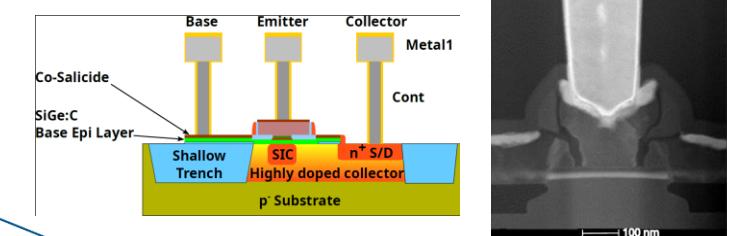


	SG13S	SG13G2	SG13G3Cu
HBT f_t/f_{max}	250 / 340 GHz	350 / 450 GHz	470 / 650 GHz
$W_{Emitter}$	170 nm	130 nm	110 nm
HBT BV_{CEO}	1.7 V	1.6 V	1.5 V
CMOS node	130 nm		
Active devices	Schottky diodes, Antenna diodes, PN diodes, ESD		
Varactors	NMOS Varactor		
Resistors	Poly-Si, Thin Film		Poly-Si
MIM Caps	1.5 fF / μm^2 (Al) 2.1 fF / μm^2 (Cu)	1.5 fF / μm^2 (Al) 2.1 fF / μm^2 (Cu)	2.1 fF / μm^2
Metallization	7 Layers AL incl. 2 & 3 μm layers or *Cu: 4 + 2 (3 μm) Al: 2 (3 μm)	7 Layers AL incl. 2 & 3 μm layers or *Cu: 4 + 2 (3 μm) Al: 2 (3 μm)	*Cu: 4 + 2 (3 μm) Al: 2 (3 μm)

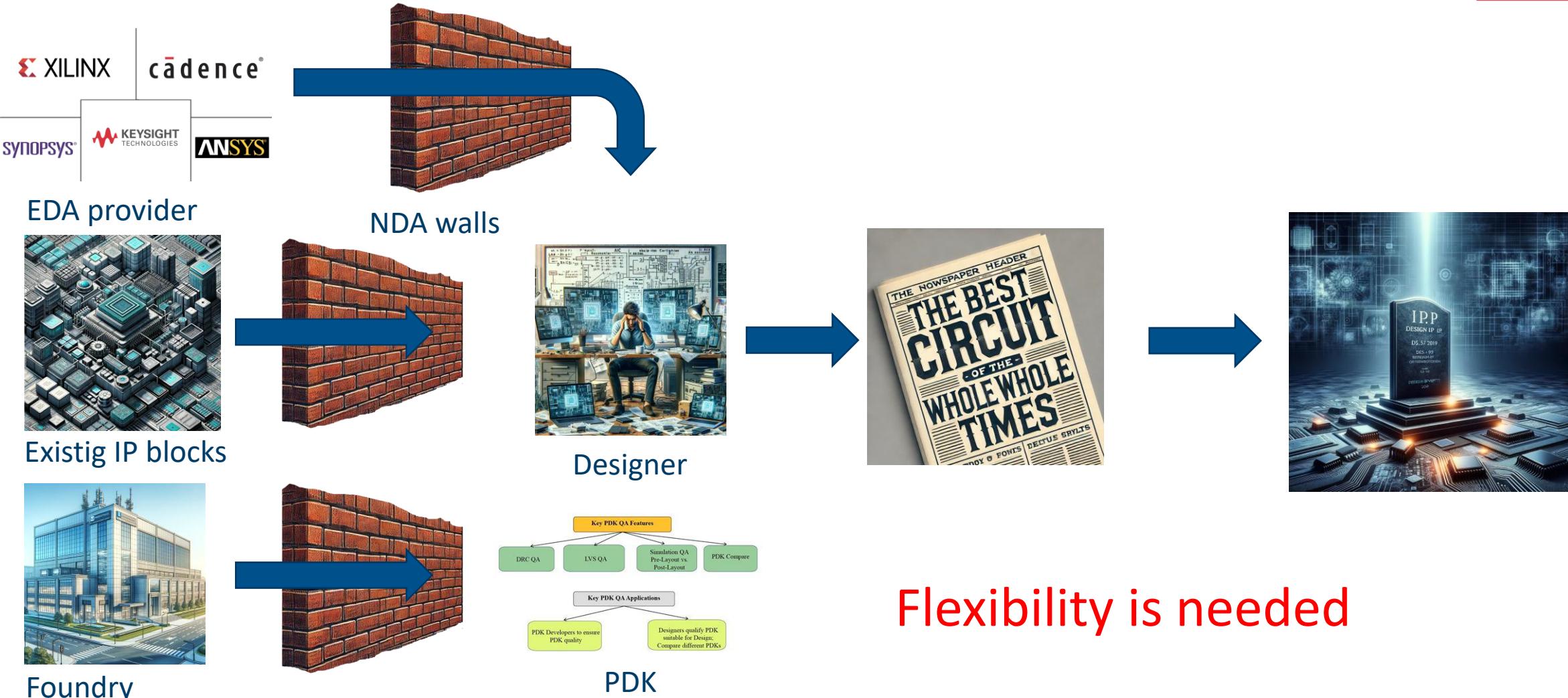
*Cu BEOL from X FAB

- SG13G2 technology was selected for the development of an open source PDK

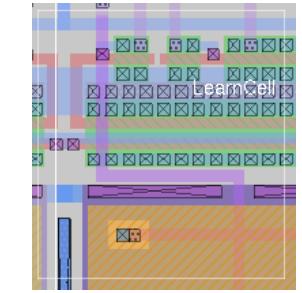
- 0 Target are high-end technology developments, low volume market introduction, technology transfer for potential mass production in commercial fabs
- 0 SG13S & SG13G2 are qualified and ready for Low Volume of high end products
- 0 SG13G3Cu is early access - qualification scheduled 2025



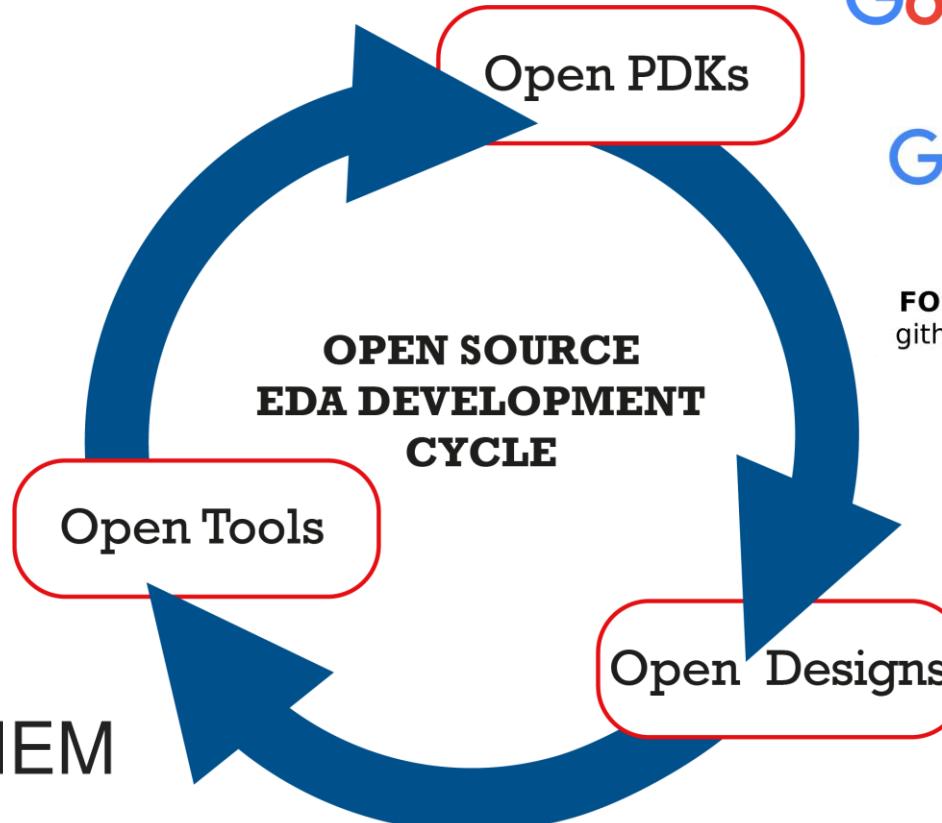
Traditional ASIC development (academic perspective)



Open source ASIC development as an alternative



XSCHEM



Google + SKYWATER TECHNOLOGY

Google + GLOBAL FOUNDRIES

FOSS 180nm Production PDK
github.com/google/gf180mcu-pdk



IHP-Open-PDK



test cases, regression tests, benchmarks, use cases,
user stories, feedback, error reports, feature requests

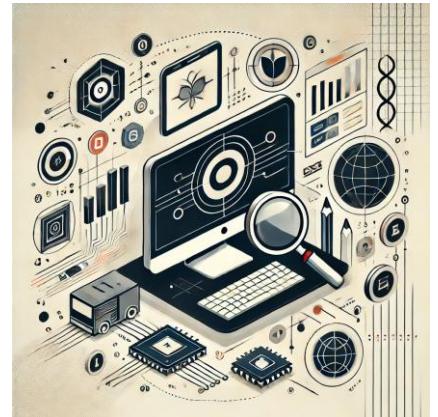
Benefits of open source ASIC design



- 0 no legal issues related to the NDA, ... anyway learn about the open source licensing (it is transparent)
- 0 you can learn a lot out of existing, already open sourced designs, reproduce it, use it as a building block of your idea
- 0 you can modify everything so there is a lot of space for research
- 0 you can still make profit.



*Open source silicon solutions are very attractive
for teaching and research*



OpenPDK Project on GitHub



The screenshot shows the GitHub repository page for IHP-Open-PDK. The repository is public and has 152 commits. It includes branches like main, ihp-sg13g2, .gitignore, AUTHORS, CODE_OF_CONDUCT.md, CONTRIBUTING.md, LICENSE, and README.md. The README file contains information about the 130nm BiCMOS Open Source PDK, its goal to provide a fully open source Process Design Kit, and its targeting of the SG13G2 process node. The repository also features an About section, releases, contributors, and languages.

IHP-Open-PDK GitHub repository, <https://github.com/IHP-GmbH/IHP-Open-PDK>, accessed: 2024-07-18.

PDK Contents:

- o Project Roadmap Gantt chart
- o Limited set of standard logic cells (Open130-G2)
- o SRAM cells
- o IO cells
- o Primitive devices (GDS)
- o KLayout:
 - o Layer property & tech files
 - o DRC deck (minimal+maximal rule sets)
 - o Filler generation
 - o LVS deck
 - o Pycells: initial API, 1st priority devices
- o ngspice/Xyce simulation models
- o xschem:
 - o device symbols, settings, testbenches
 - o stdcells symbols
- o Qucs-S symbols and examples
- o OpenEMS: tutorials, scripts, documentation
- o SG13G2 Process Specification & Layout Rules
- o MOS/HBT/Passive device measurement data
- o GDS QA Test structures for DRC/LVS

Working the IHP Open PDK



Welcome to IHP 130nm BiCMOS Open Source PDK documentation!



This documentation is currently a work in progress.



Current Status – Experimental Preview



Search docs

- PDK Contents
- Installation
- Process Specifications
- Layout Rules
- Analog Design
- Digital Design
- Physical & Design Verification
- Contribution
- References



1. Dependencies

The tools supported by IHP-Open-PDK are open source and are not always distributed as binaries or through packages available to install using programs such as apt-get. In order to use the tools one have to compile/build it from the sourc code usually available on platforms like github, gitlab, sourcforge codeberg. Having all the build tools installed and meeting all necessary dependencies the installation program is usually straightforward.

1.1. Build tools

The first step to build a tool/program from a source code is to have build tools, what means necesary compilers and make systems, which allows the user to build the source code.

```
sudo apt-get install -y build-essential  
sudo apt-get install -y qtbase5-dev qttools5-dev  
sudo apt-get install -y clang cmake libtcltool autoconf  
sudo apt-get install -y python3 python3-dev python3-pip python3-virtualenv python3-venv  
sudo apt-get install -y ruby ruby-dev
```

1.2. Useful tools

Before performing installation from sources it is recommended to install some tools that are useful:

```
sudo apt-get install -y btop tree xterm graphviz git  
sudo apt-get install -y octave liboctave-dev
```

Read the docs: <https://ihp-open-pdk-docs.readthedocs.io/en/latest/index.html>

Some of the Information Available in ChatGPT



1. Process Technology

- Overview of advanced semiconductor technology, including high-performance devices and integrated components
- Offers various modules for expanded functionality

2. Multi-Project Wafer (MPW) Services

- Includes details on pricing, schedules, and supported processes
- Specifies chip area requirements and approval process for smaller designs

3. Layout and Process Specifications

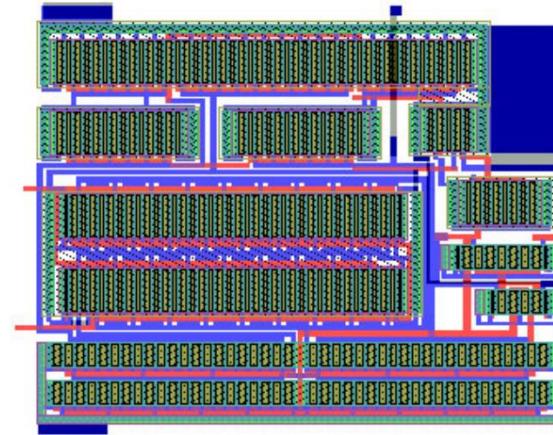
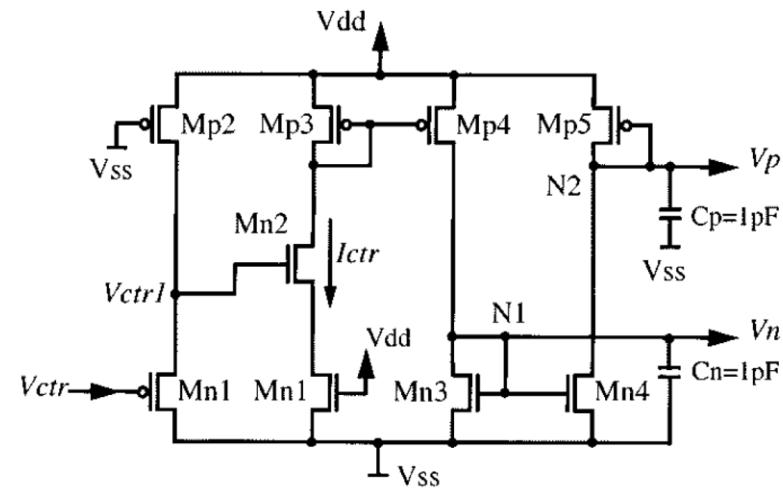
- Defines design rules, device specifications, and physical constraints
- Includes details on available materials and process layers



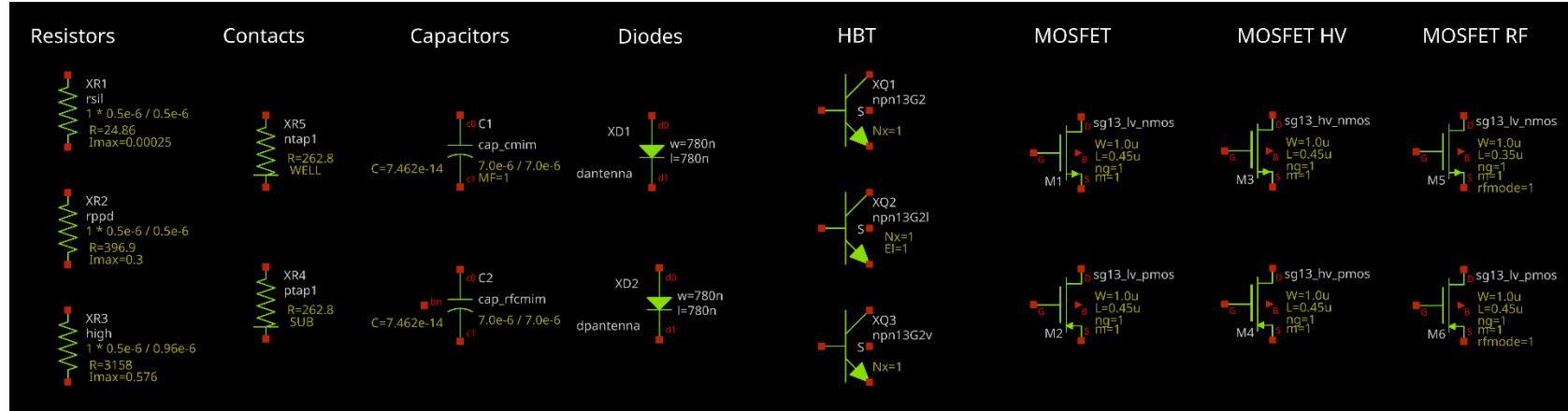
Some open source EDA tools – analog/mixed/RF



Schematic	Simulators	Layout	DRC	LVS	PEX	EM
Xschem	Ngspice	Klayout	Klayout	Klayout	Magic	FastCap
Qucs-S	Xyce	Magic	Magic	netgen		OpenEMS
	gnucap	GDSFactory				Elmer



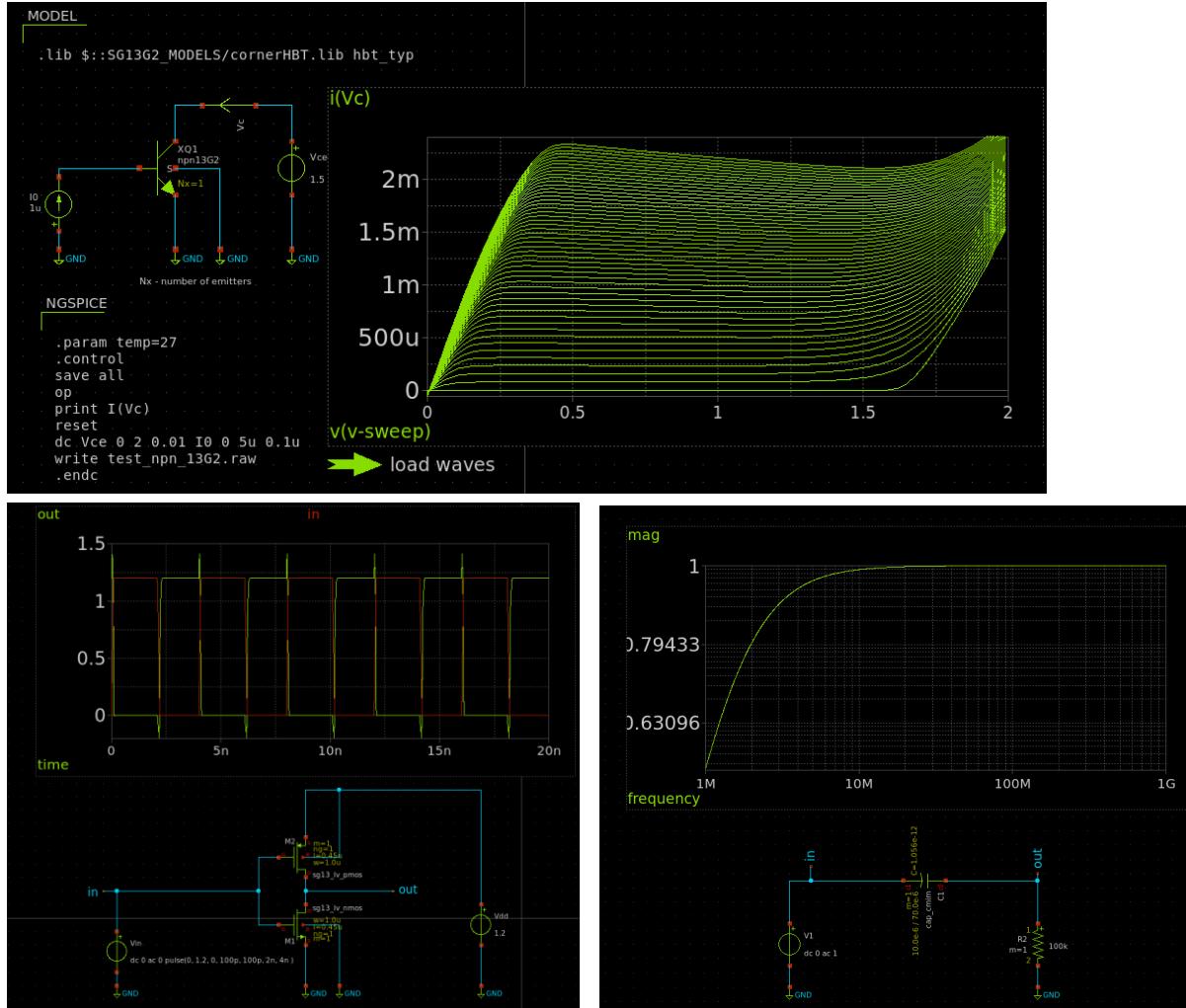
OpenPDK support for schematic capture



The current version of the IHP OpenPDK supports:

- xschem primitives for schematic capture
- automatic ngspice/Xyce compatible netlist generation
- example use cases to show the basic functionalities and parameters of the primitives

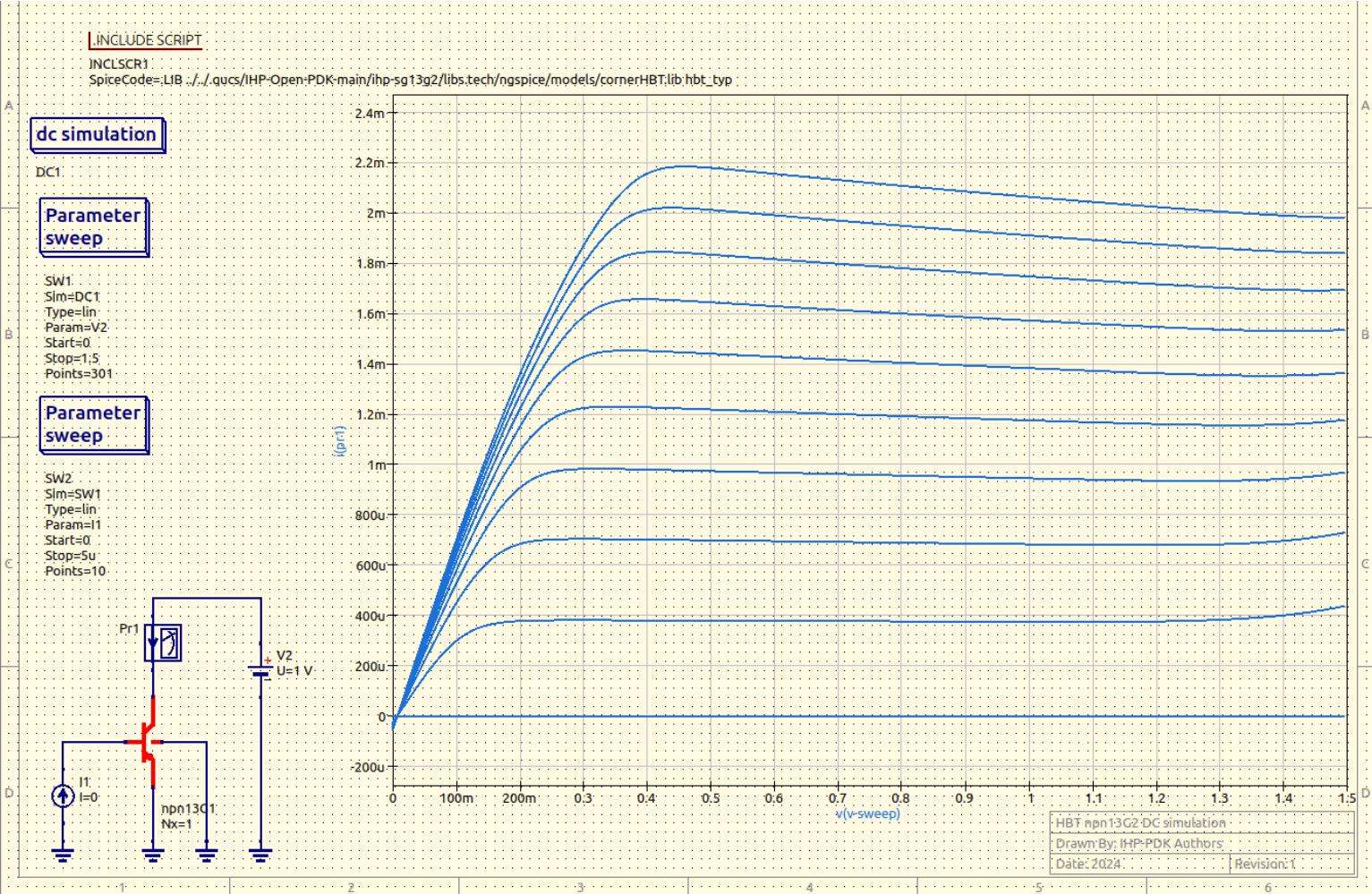
OpenPDK support for simulations



Analog simulation facilities:

- PSP103.6 MOSFET models from SemiMod,
- ngspice/Xyce compatible netlists,
- process corners: typical, best case, worst case,
- statistics related to the process variation,
- simulation examples segmented by simulation type: DC, TRAN, AC, MonteCarlo, S-parameters
- postprocessing python scripts,
- model extensions for RF frequency range,
- ngspice 40+ compatible

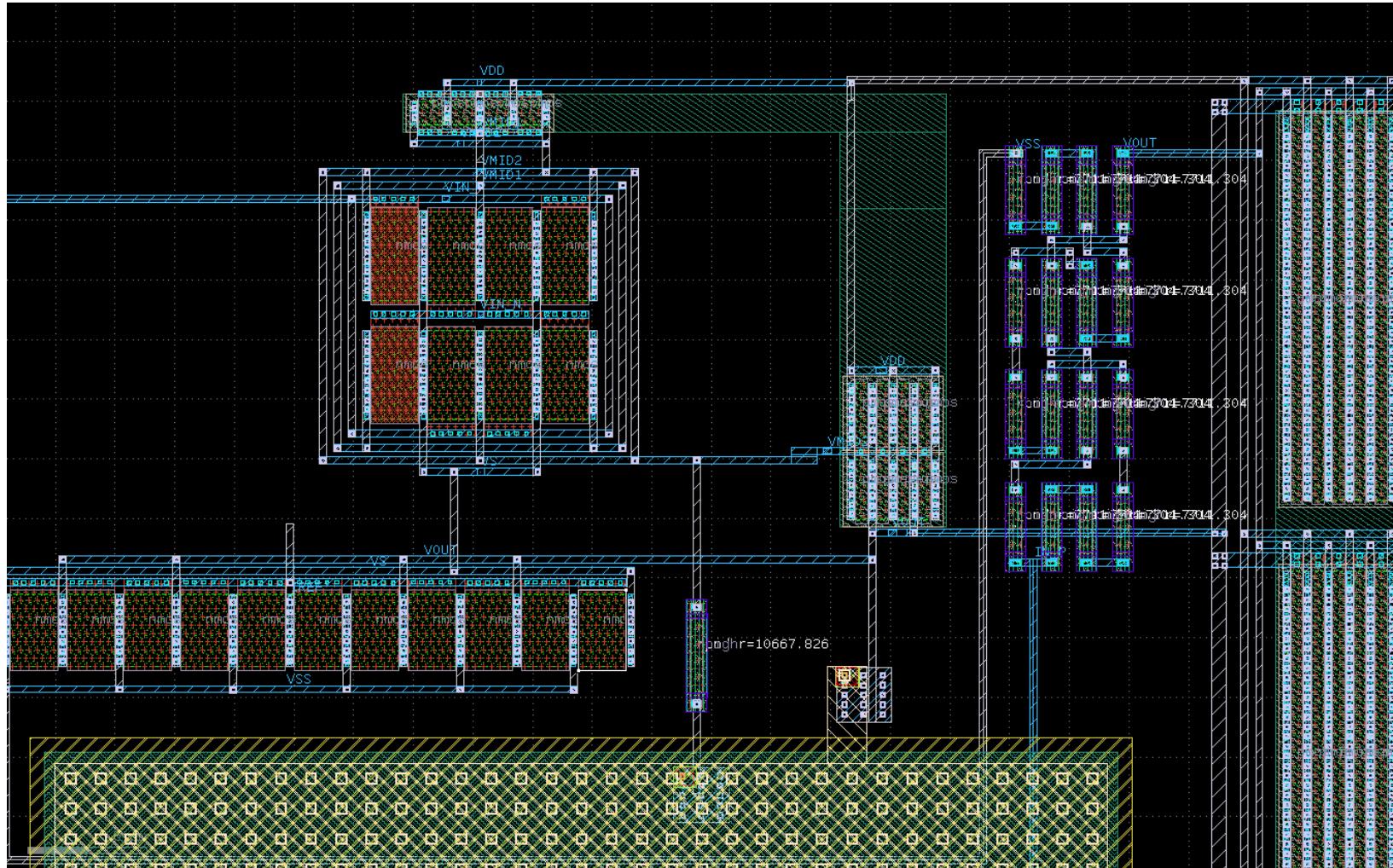
OpenPDK support for Qucs-S schematic capture



The current version of the IHP OpenPDK supports:

- 0 primitives for schematic capture
- 0 automatic ngspice/Xyce compatible netlist generation
- 0 example use cases to show the basic functionalities and parameters of the primitives
- 0 XSPICE model support (under development)

KLayout – primary tool for open source layout design

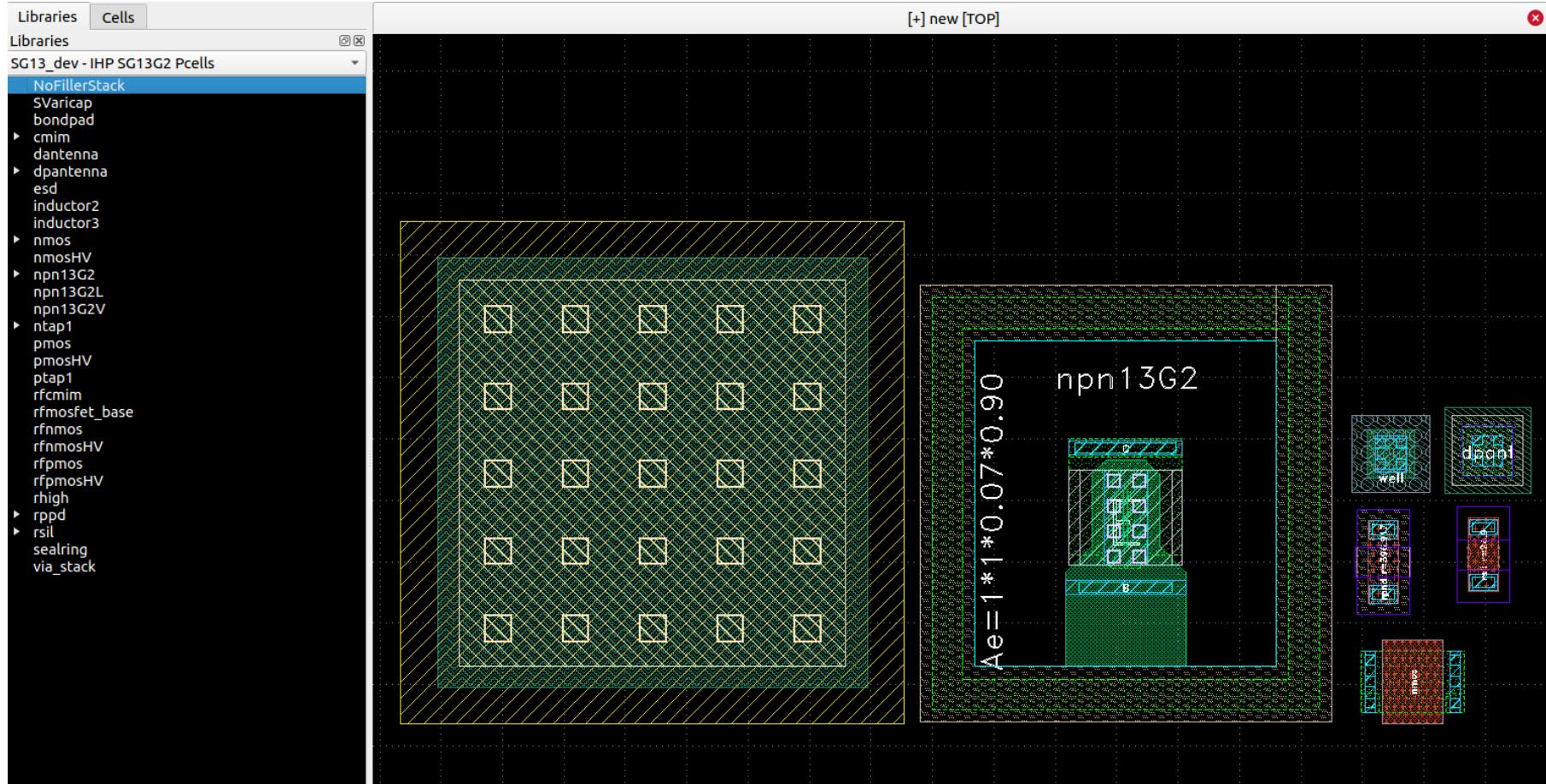


KLayout key features:

- o hierarchical view
- o parametric cell support
- o DRC/LVS checks
- o command line batch mode
- o XOR and DIFF tools
- o Custom scripts in Python/Ruby
- o Plugins

<- fragment from LDO design ([link](#))

KLayout – Pycell support



- 0 Pycells compatible with Synopsys PyCell Studio and Keysight ADS
- 0 KLayout wrapper API development

KLayout example DRC run on `gatpoly` QA cell



The screenshot shows the KLayout 0.28.12 interface with a project named "sg13g2_qacells.gds [gatpoly]".

Cells: gatpoly, metal1, nwell.

Layers: Activ.drw, GatPoly.drw, GatPoly.flr, Cont.drw, Metal1.drw, Metal1.pin, pSD.drw, NWell.drw, Substrate.drw, ThickGateOx.drw, TEXT.drw.

Marker Database Browser:

Cell / Category	Count (Not Visited)
By Cell	37 (35)
[gatpoly]	37 (35)
aFil.g	1
aFil.g2	4 (4)
GFil.g	1
Gat.d	8 (8)
M1.j	1 (1)
M1Fil.h	4 (4)
Gat.a	12 (12)
Gat.b	6 (6)
By Category	37 (35)
All	37 (35)

Info: Min. GatPoly to Activ space = 0.07

KLayout example LVS run on sg13_lv_nmos mosfets



Netlist LVS

... on layout sg13_lv_nmos.gds

Netlist Schematic Cross Reference Log

Circuits Objects Layout Reference

sg13_lv_r sg13_lv_nmos ↔ \$ sg13_lv_nmos SG13_LV_NMOS

► -▷ Pins
► ↑ Nets
▼ ↴ Devices

- ↴ sg13_lv_nn \$11 / sg13_lv_nmos [L=(N1 / SG13_LV_NMOS [L=0.13, W=0.15]
- ↴ sg13_lv_nn \$12 / sg13_lv_nmos [L=(N2 / SG13_LV_NMOS [L=0.13, W=0.2]
- ↴ sg13_lv_nn \$14 / sg13_lv_nmos [L=(N3 / SG13_LV_NMOS [L=0.15, W=0.2]
- ↴ sg13_lv_nn \$9 / sg13_lv_nmos [L=0. N4 / SG13_LV_NMOS [L=0.15, W=0.3]
- ↴ sg13_lv_nn \$6 / sg13_lv_nmos [L=0. N5 / SG13_LV_NMOS [L=0.3, W=0.3]
- ↴ sg13_lv_nn \$13 / sg13_lv_nmos [L=(N6 / SG13_LV_NMOS [L=0.25, W=0.6]
 - -○ S ↔ D ⚠ \$35 (1) D6 (2)
 - -○ D ↔ S ⚠ \$36 (1) S6 (2)
 - -○ G \$37 (1) G6 (2)
 - -○ B \$1 (26) SUB (27)
- ↴ sg13_lv_nn \$16 / sg13_lv_nmos [L=(N7 / SG13_LV_NMOS [L=0.15, W=0.6]
- ↴ sg13_lv_nn \$4 / sg13_lv_nmos [L=3. _PATTERN_37 / SG13_LV_NMOS [L=3.74, W=5.55]
- ↴ sg13_lv_nn \$5 / sg13_lv_nmos [L=4. _PATTERN_40 / SG13_LV_NMOS [L=4.6, W=7.09]

KLayout LVS ruledeck:

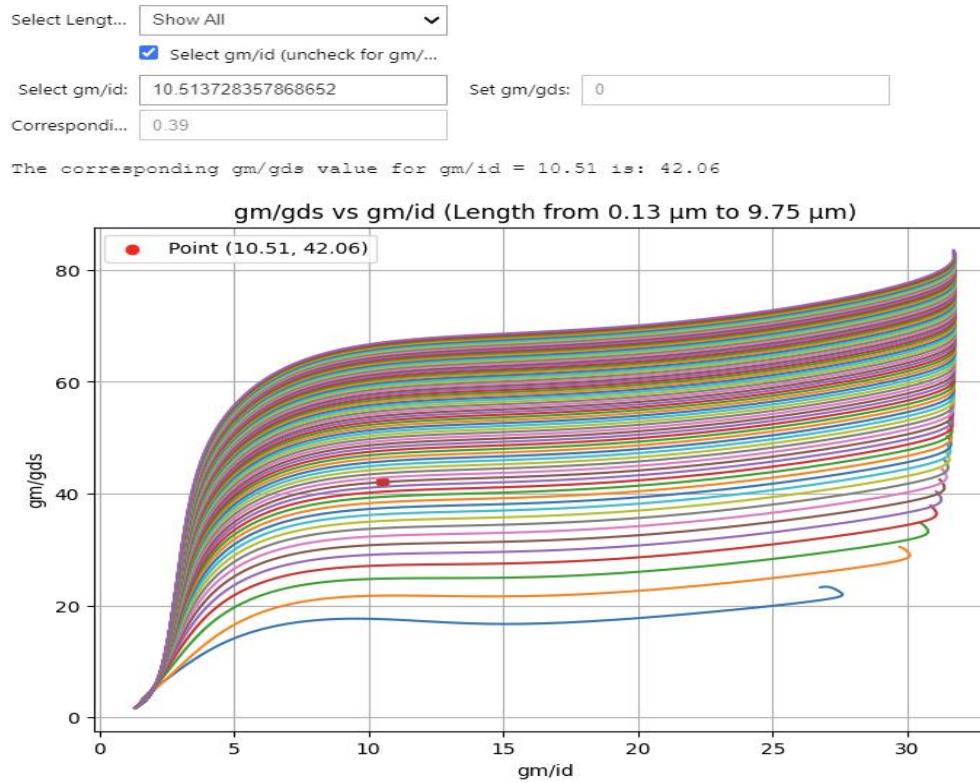
- o accepts CDL netlist,
- o extracts devices from layout,
- o creates extracted netlist,
- o performs checks and compares:
 - o Pins
 - o Nets
 - o Devices
- o reports inconsistencies
- o can run in batch mode

OS tools for Advanced IC Design?



MOSFET Sizing with GM/ID Curves:

- 0 Using Python scripting alongside Ngspice to generate GM/ID curves for efficient MOSFET sizing



Mixed Signal Design

- 0 Verilator
- 0 Xspice
- 0 Creating digital models in conjunction with analog design

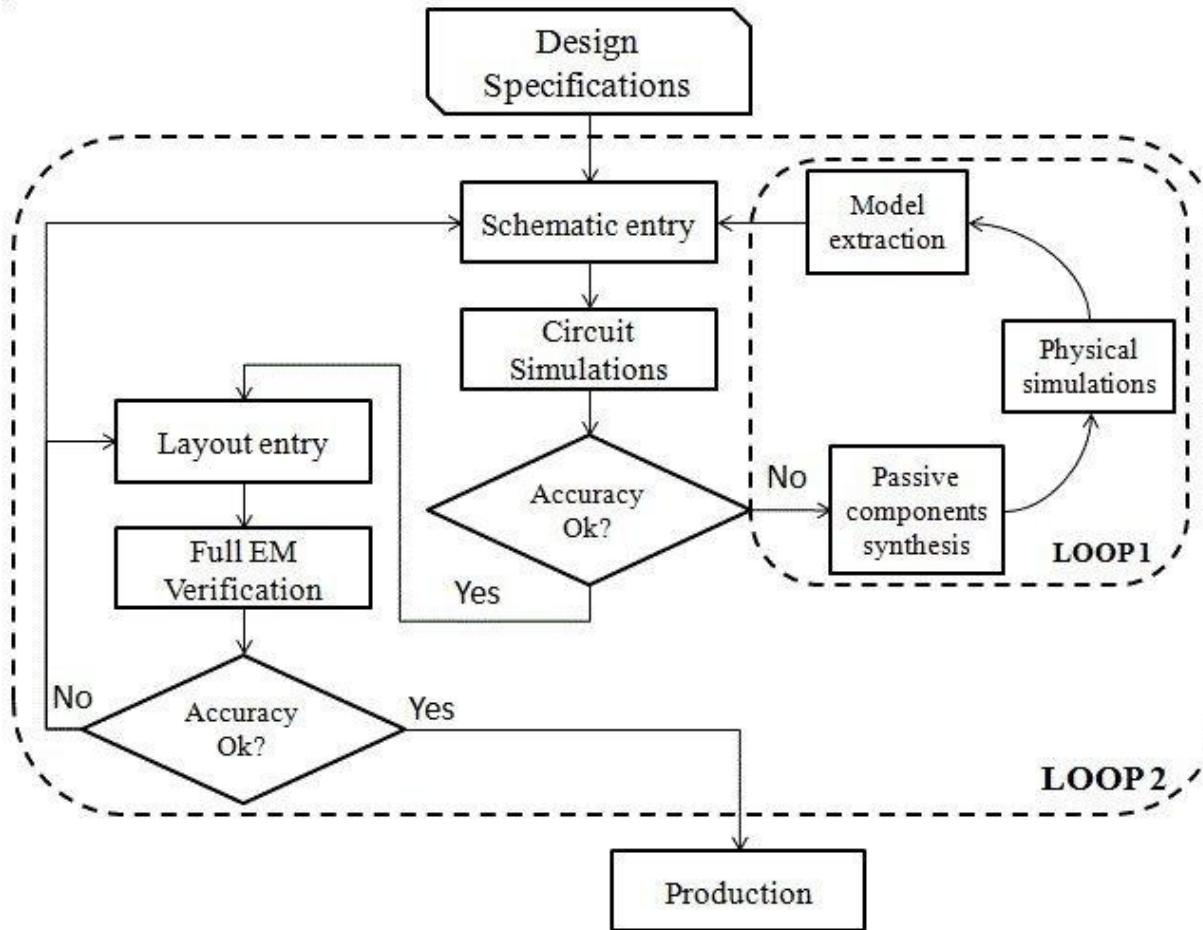
Layout Automation

- 0 Physical verification, filler scripts
- 0 Pcells generation from SPICE-Files
- 0 Streamlining the layout process

What about the Tools for Radio Frequency Design?



Streamlining The Process of Design



- o Repetition highlights the need for efficiency!
 - o We need integration of tools
 - o Ongoing work at IHP!

Ahyoune, S., Sieiro, J., Lopez Villegas, J. M., Vidal, N., Carrasco, T., Ramos, F., Fernández-Sanjuán, J., & Albero, F.-F. (2013). *Scalable LTCC library for System-in-Package design*.

Interoperability from Open Source to Closed Source



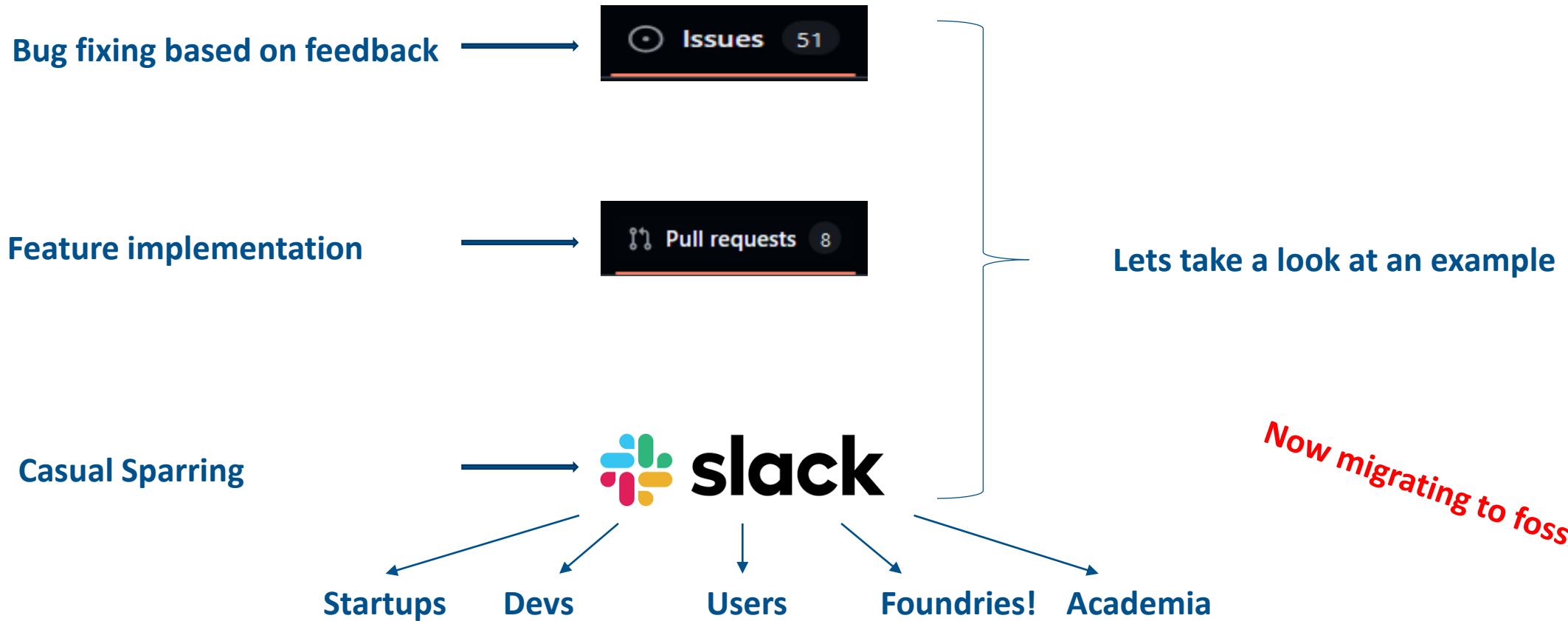
Schematic/Simulation

- 0 Not possible since it would require common database

Layout

- 0 Definitely possible to handle GDS files
- 0 Possibility of handling physical verification with commercial tools
- 0 Possibility of handling post process simulation with commercial tools

How Does the OS Community Aid the Development



Example on How to use the OS community



Opening an issue ticket on Github



Next steps?

Opening a pull request on Github



Next steps?

Doubts and Fears regarding the tools



```
# ihp-sg13g2  
# xschem  
# klayout
```



Enjoy!

Work in progress – analog/mixed/RF flow



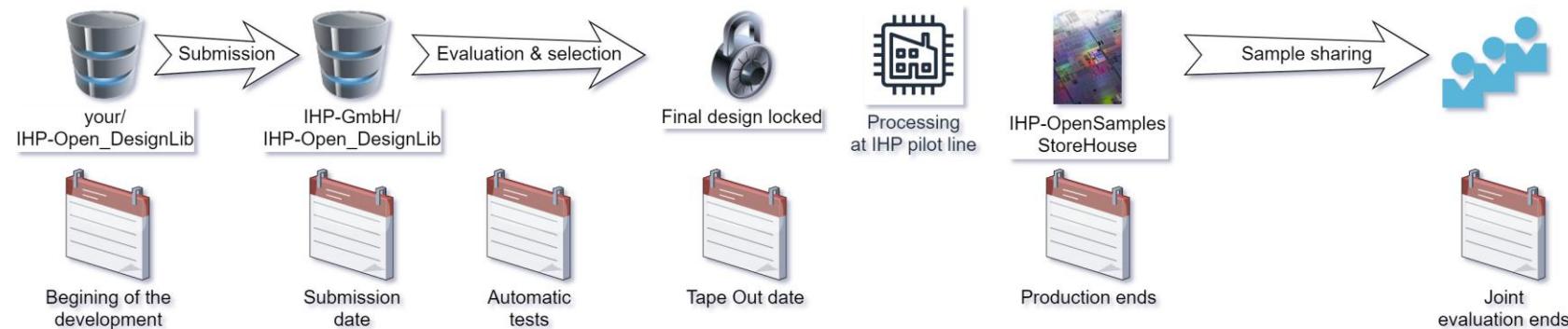
- 0 Parasitics Extraction PEX – ongoing
- 0 Noise modeling in ngspice (transient noise, low frequency noise) – ongoing
- 0 Qucs-S support - ongoing
- 0 Device models – issues found and reported by community members!
- 0 Even more models to come! – (rfcmim, rfpmos, Svaricap etc...)
- 0 DRC – a 80% subset of rules available
- 0 Klayout PyCells and PyCell API - ongoing
- 0 OpenEMS integration for EM field solving (**We will use this**)
- 0 Mixed mode testcases using xspice + verilator (**We will use this**)

Free MPW Runs - support open source PDK & design

- 0 The table provides schedule of MPW Runs for FMD-QNC project in 2024 and 2025

Tape out date	22/05/24	11/11/24	22/11/24	01/03/25	09/05/25	18/07/25	15/09/25
Technology	SG13G2	SG13CMOS	SG13G2	SG13G2	SG13G2	SG13G2	SG13CMOS
Area [mm ²]	10	220	20	140	30	30	220

For more details check: <https://ihp-open-ip.readthedocs.io/en/latest/>



- 0 Project funds can be used exclusively to produce chip designs for non-economic activities, such as university education, research projects, and others
- 0 A concept for sustainable provision of free or low-cost MPW area for the open source community is to be developed.

Criteria for design IP selection from open community

Mandatory criteria for IP selection

- Completeness of IP data (all data need to be open source)
- DRC error free designs
- Area below 2 mm² preferred (larger designs only if area is available)
- Potential export restrictions

Additional criteria for IP selection

- First time submission (preferred)
- Design should use open source tools supported by IHP open PDK
- For SG13G2 runs designs using SiGe (preferred)
- Documentation quality
- Uniqueness, not yet seen designs (i.e. if there were no ADCs before, an ADC design would get a higher point)



Flow to upload designs for MPW run

- 0 Before a design can be considered for fabrication start a pull request at https://github.com/IHP-GmbH/TO_<date>
- 0 All design data need to be submitted at **least 2 weeks** before TAPE OUT
- 0 Europractice will check designs regarding mandatory criteria (Completeness, DRC clean GDS, ...)
- 0 Europractice registers area and upload selected GDS files
- 0 Designs will be processed by IHP
- 0 IHP can rent samples for joint evaluation **under certain agreement (Continuation on next slide)**
- 0 Evaluation results need to be published on <https://github.com/IHP-GmbH/IHP-Open-DesignLib>
- 0 Evaluation to provide permanent samples are under evaluation

The principal goal is to build an open source design library for future projects

Joint Evaluation In Detail

- 0 Its all bound on legal reasons!
- 0 Evaluation can be made for about 2 years
- 0 Joint evaluation allows your chip to be tested across different systems and equipment, providing valuable feedback to enhance your design





How to reach us

Email

- 0 openpdk@ihp-microelectronics.com

GitHub

- 0 issues – for reporting issues, questions

- 0 discussions – for requesting features

Fossi-Chat

- 0 general discussions

- 0 announcements

The screenshot shows the GitHub repository interface for 'IHP-GmbH / IHP-Open-PDK'. At the top, there are tabs for 'Code', 'Issues' (24), 'Pull requests' (4), and 'Discussions'. Below the tabs, the repository name 'IHP-GmbH / IHP-Open-PDK' is displayed. On the left, there's a sidebar with 'Home', 'People', 'Rooms', and the 'general' room selected. The main area shows a list of messages in the 'general' channel. Some messages include:

- @nbharath:fossi-chat.org and 4 others joined (Sat, Mar 29, 2025)
- @manigandan:fossi-chat.org joined the room (Sun, Mar 30, 2025)
- Mathias DL9MJ changed their display name to Matthias F/DL9MJ (Mon, Mar 31, 2025)
- M @mballance:fossi-chat.org and 4 others joined, @seb.yang:fossi-chat.org joined and changed their profile picture (Mon, Mar 31, 2025)
- Matt Venn please sign this letter! <https://open-source-chips.eu/> (Tue, Apr 1, 2025)
Open Letter: Open-Source Chips for Europe
Open Source Chips for Europe Open Letter on the Urgency of Access to Open Source Chip Manufacturing To Who set ambitious goals and its implementation is a significant pan-european effort. From an academic perspective, la...
- @kchen:fossi-chat.org joined the room (Tue, Apr 1, 2025)
- @daerthjwb:fossi-chat.org and 5 others joined, @liprin:fossi-chat.org joined and left (Wed, Apr 2, 2025)
- @vikas:fossi-chat.org and 3 others joined, @donna:fossi-chat.org removed a message, larry_harris left (Thu, Apr 3, 2025)
- @jazim_ibrahim:fossi-chat.org and 7 others joined, Mathias DL9MJ changed their name (Thu, Apr 3, 2025)
- Reza Papi Hello everyone, does anyone have suggestions or a recommended setup for measuring the PSRR of an ASIC amplifi... (Fri, Apr 4, 2025)
8 replies Reza Papi Good to know that 😊 Thank you Sylvain
- @harryxni:fossi-chat.org joined the room (Fri, Apr 4, 2025)

Laying the Groundwork: A Soft Start to Build a Strong Foundation

Repository reference:
modules/modules_0_foundations

`grip module_0_foundations/foundations.md`

```
* Serving Flask app 'grip.app'  
* Debug mode: off  
WARNING: This is a development server. Do not use it in a production deployment.  
Use a production WSGI server instead.  
* Running on http://localhost:6419  
Press CTRL+C to quit
```

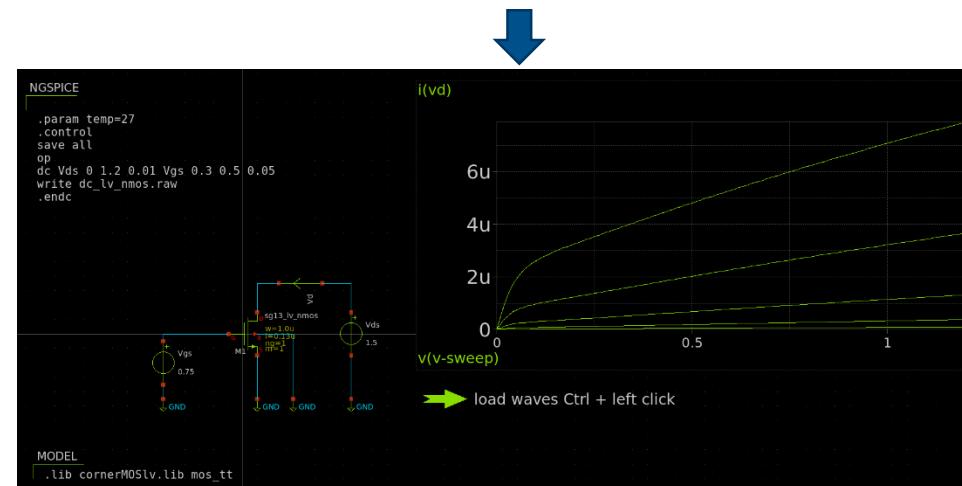
Ctrl + left click

Launching Xschem

- With the IHP Open PDK setup, launch Xschem by simply typing “xschem” in the console

- This opens the Open PDK example library for Xschem. Select an example and press E to descend into it. Use **Ctrl+E** to go back.

- Click **Netlist**, then **Simulate** to start the simulation. Once completed, view the results by **Ctrl + Left Click** on the green arrow.



Ngspice Code Block: Defines the simulation setup



- 0 **.controlendc:** Specifies a sequence of commands to control the simulation
- 0 **op:** Executes a DC operating point analysis
- 0 **Save all:** Saves all operating points, node voltages, currents, etc., for later review in the Xschem terminal
- 0 **The dc Vds:** Performs a DC parametric sweep of **Vds** from 0V to 1.2V in 0.01V steps (and similarly for other parameters)
- 0 **Write:** Saves the final DC results to a file, **dc_lv_nmos.raw**, in the simulation directory (auto generated folder)

```
NGSPICE
.param temp=27
.control
save all
op
dc Vds 0 1.2 0.01 Vgs 0.3 0.5 0.05
write dc_lv_nmos.raw
.endc
```

Additional Setup

- 0 **Model:** Specifies which model we want to use. In this case we are using the low voltage transistor in the typical-typical corner

MODEL

```
.lib cornerMOSlv.lib mos_tt
```

- 0 **Waveform Reload Launcher:** This is where we define the path to the data that we want to plot in the diagram in the schematic, more on this later..

→ load waves Ctrl + left click

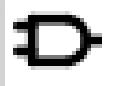
```
name=h5  
descr="load waves Ctrl + left click"  
tclcommand="xschem raw_read $netlist_dir/dc_lv_nmos.raw dc"
```

Now Its Your Turn (Creating the first schematic)

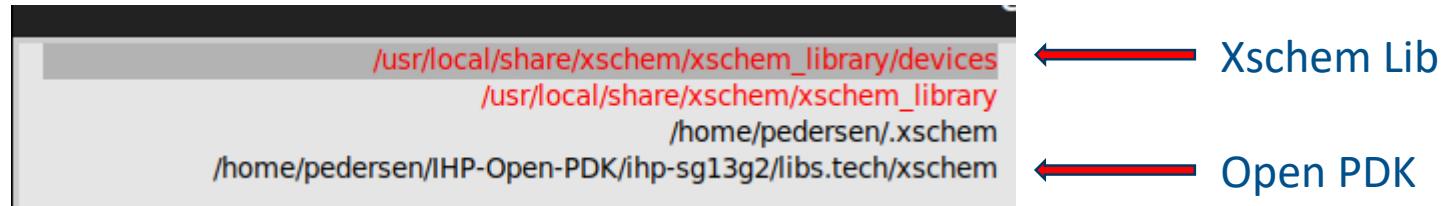


Stay Organized! Before we begin, we need to create a **.sch** file for our schematic. Navigate to your desired location and create the schematic by running:

```
touch test.sch
```

Open Schematic: Open the schematic, then navigate to
Components  (Ctrl + i)

Here, you'll find both the **Xschem library** and the **Open PDK library**.



Placing Components

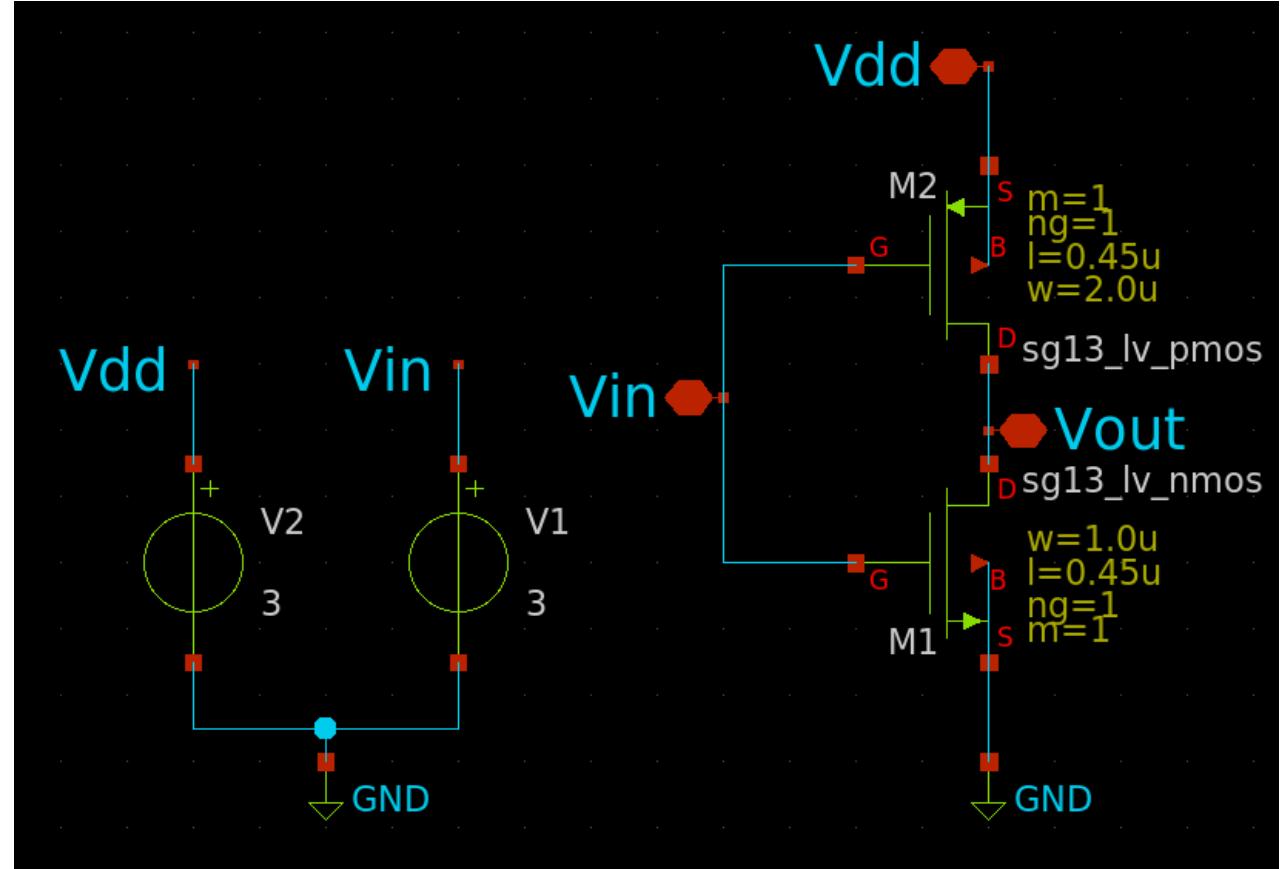


List of components:

- 0 xschem_library/devices: vsource.sym (x2)
- 0 xschem_library/devices: gnd.sym (x2)
- 0 xschem_library/devices: lab.sym (x2)
- 0 xschem_library/devices: iopin.sym (x3)
- 0 libs.tech/xschem/sg13g2_pr: sg13_lv_nmos
- 0 libs.tech/xschem/sg13g2_pr: sg13_lv_pmos

Remember to double the width of the PMOS!

- 0 To wire components press **W**
- 0 To center the schematic press **F**
- 0 To rotate press **(shift + R)** to flip press **(shift + F)**





Viewing The Netlist

Being comfortable with netlist viewing is essential, as it reveals how the simulator interprets your circuit.

To extract the netlist:

- 0 Open Options and navigate to **Netlist Format / Symbol Mode**.
- 0 Select **SPICE Netlist** to ensure compatibility with Ngspice.
- 0 Go to the **Simulation** tab and click **Show Netlist After Netlist Command (Shift + A)**.

```
** sch_path: /home/pedersen/projects/IHP-AnalogAcademy/module:  
**.subckt inverter Vout Vdd Vin  
.iopin Vout  
.iopin Vdd  
.iopin Vin  
V1 Vin GND 3  
V2 Vdd GND 3  
XM1 Vout Vin GND GND sg13_lv_nmos w=1.0u l=0.45u ng=1 m=1  
XM2 Vout Vin Vdd Vdd sg13_lv_pmos w=2.0u l=0.45u ng=1 m=1  
**.ends  
.GLOBAL GND  
.end
```

Viewing The Netlist

- 0 First Section indicates the subcircuit which is commented out in this case since we are looking at a standalone simulation

```
**.subckt inverter Vout Vdd Vin  
.iopin Vout  
.iopin Vdd  
.iopin Vin
```

- 0 Next section defines the voltage sources in this **Vin** is set to 3V relative to **Gnd** and same for **Vdd**

```
V1 Vin GND 3  
V2 Vdd GND 3
```

- 0 **Transistor definition:** M<XName> <Drain> <Gate> <Source> <Bulk> <Model> W=<Width> L=<Length> ng=<number of gates> m=<multiplier>

```
XM1 Vout Vin GND GND sg13_lv_nmos w=1.0u l=0.45u ng=1 m=1  
XM2 Vout Vin Vdd Vdd sg13_lv_pmos w=2.0u l=0.45u ng=1 m=1
```

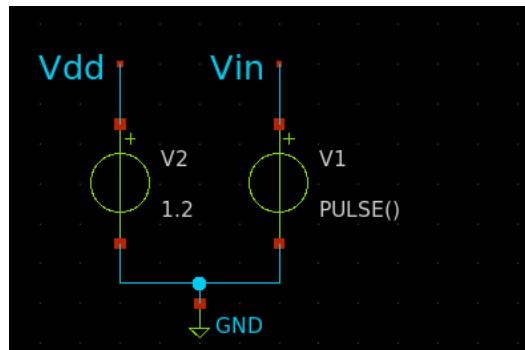
- 0 **.Global GND:** As the name indicates it's a global node reference

```
.GLOBAL GND
```

Complete the Testbench Using These Hints

- 0 Xschem_library/Devices: code_shown.sym (x2)
(Hint: Check examples in the PDK by running Xschem in the terminal)

- 0 Vdd/Logic High: 1.2V



MODEL
.lib

NGSPICE
.control
save all
write test_inverter.raw
.endc

Something is missing!

4.1.1 Pulse

General form:

PULSE(V1 V2 TD TR TF PW PER NP)

11.3.10 .TRAN: Transient Analysis

General form:

.tran tstep tstop <tstart <tmax>> <uic>

Syntax



code_shown.sym



```
name=MODEL only_toplevel=true  
format="tcllevel( @value )"  
value="  
.lib cornerMOSlv.lib mos_tt  
"
```



```
MODEL  
.lib cornerMOSlv.lib mos_tt
```

code_shown.sym



```
name=NGSPICE  
only_toplevel=true  
value="  
.control  
save all  
tran 50n 2u  
write test_inverter.raw  
.endc  
"
```

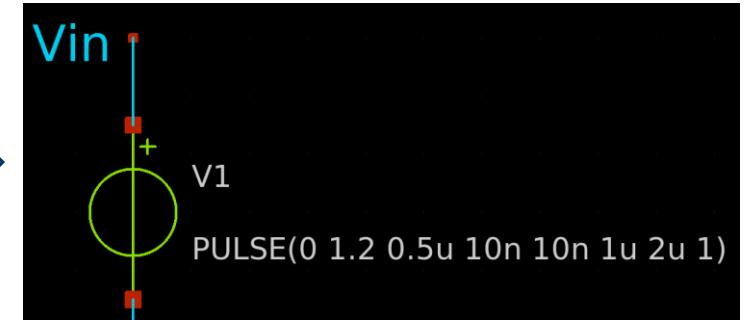


```
NGSPICE  
.control  
save all  
tran 50n 2u  
write test_inverter.raw  
.endc
```

vsource.sym



```
name=V1  
value="PULSE(0 1.2 0.5u 10n 10n 1u 2u 1)"  
savecurrent=false
```



Final Schematic

Commands in Xschem Terminal:

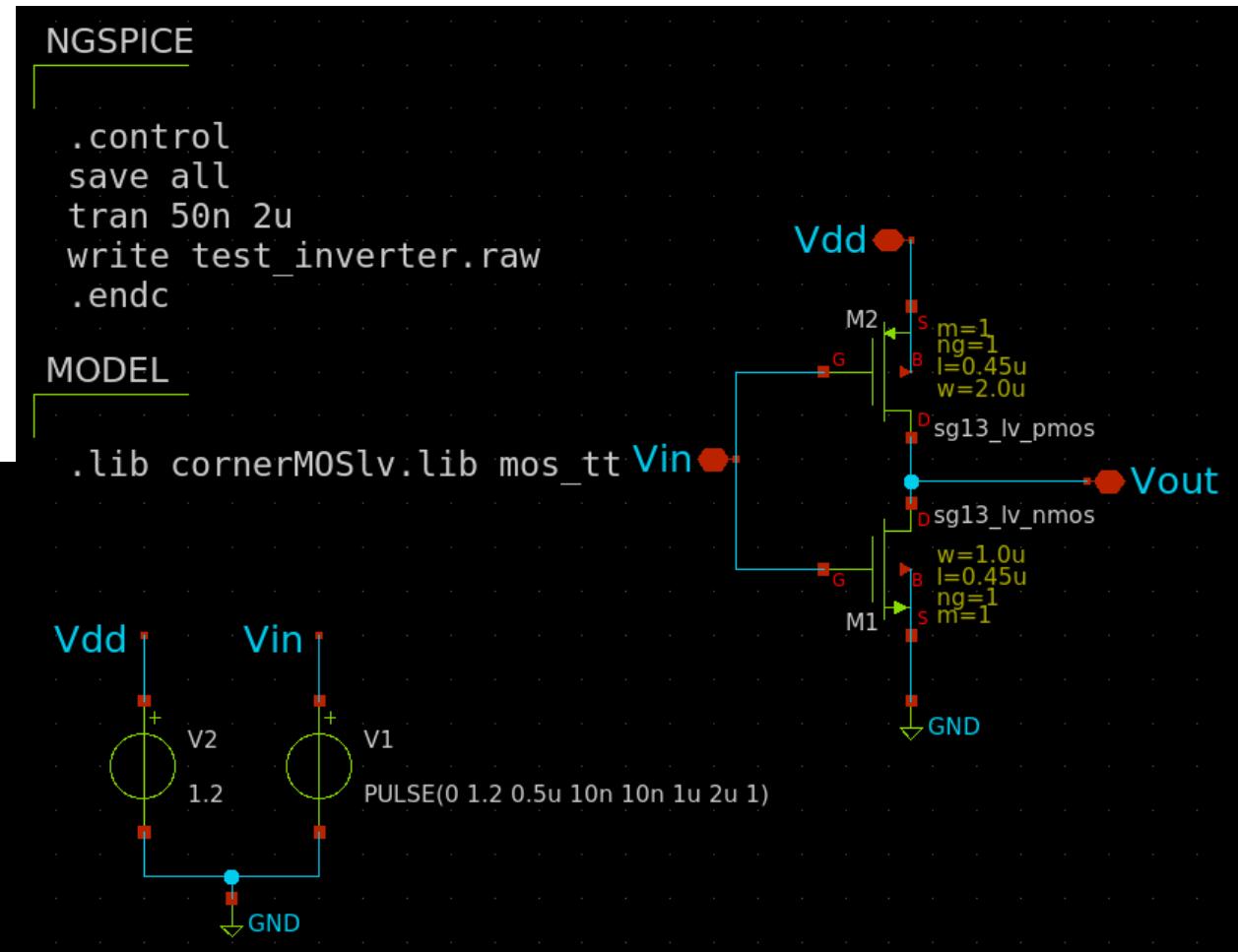
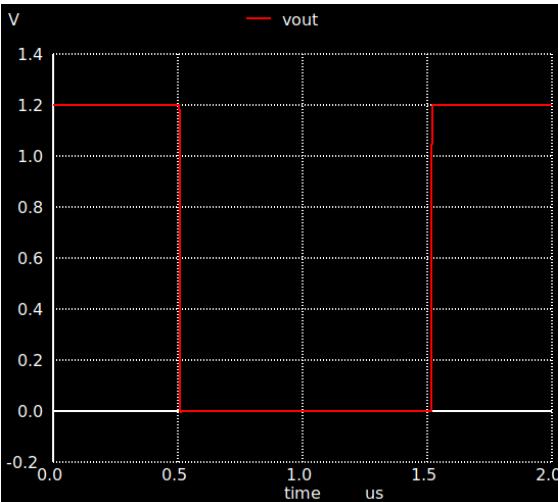
- o show all (shows for example transistor small signal parameters)
- o display (shows available data)
- o Plot (plots data ☺try with plot vout)

```

Doing analysis at TEMP = 27.000000 and TNOM = 27.000000
Using SPARSE 1.3 as Direct Linear Solver
Initial Transient Solution
-----
Node          Voltage
-----
vin          0
vdd          1.2
vout         1.2
v2#branch   -3.09018e-10
v1#branch   8.18116e-11

No. of Data Rows : 80
binary raw file "test_inverter.raw"
ngspice 74 -> ■

```



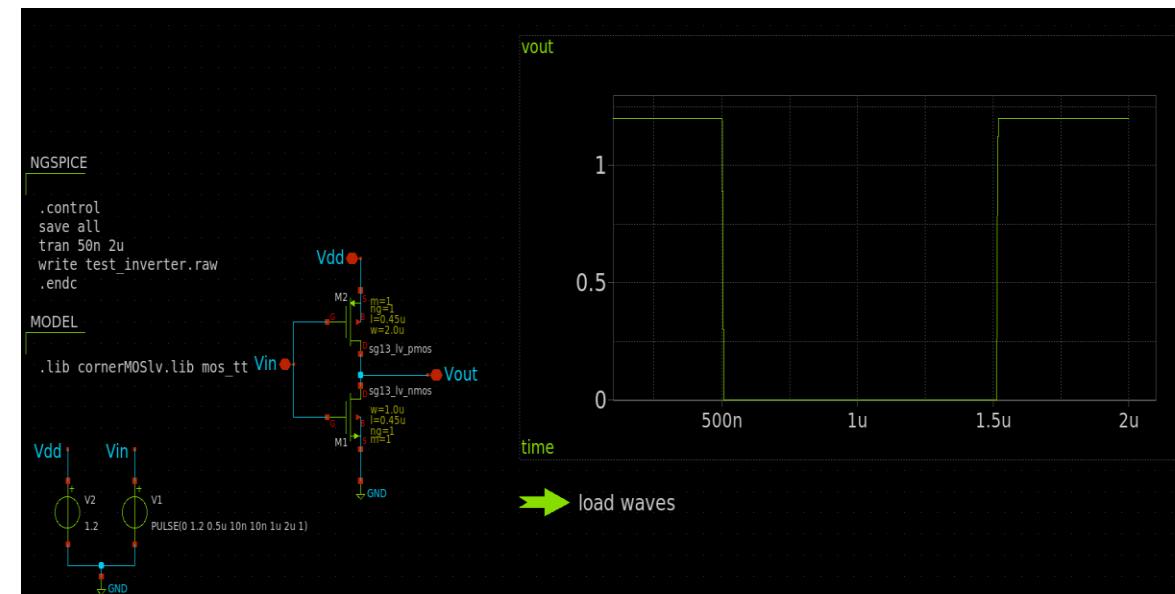
Setting A Fixed Plot In Schematic

Lets review the simulation outcome:

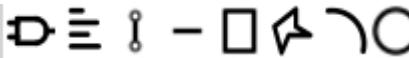
- 0 Navigate to the directory of the schematic. You should see a simulation folder.
- 0 Inside this folder you should find the **.raw** file and the **.spice** file. We will use the **.raw** file to set the plot since this contains our simulation data

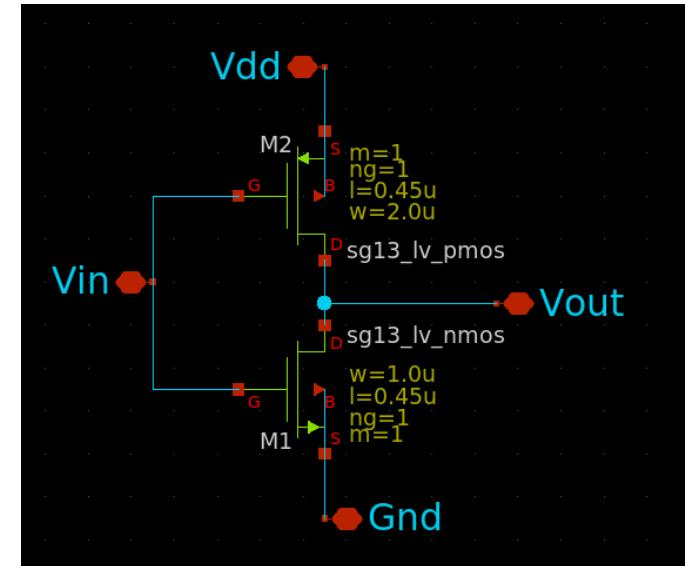
Next insert the plot in the schematic:

- 0 Navigate to the **Simulation** tab and under **graphs** press **add waveform graph** and **add waveform reload launcher**
- 0 Double click the launcher and make sure you point to right **.raw** file
- 0 After this load the waves (**Ctrl + Left click**) and double click on the wave form viewer. Now select the data to be plotted
- 0 By pressing **F** on both axes you can scale the plot



Creating a Symbol

- 0 The simplest way to create a symbol (from the testbench) is by duplicating the existing schematic and renaming the copy to **inverter_tb.sch**
The original schematic can then be used to define the symbol subcircuit
- 0 Once this is done, we can delete everything except for the inverter (Remember to put IO pin on the source of the NMOS)
- 0 For Creating the Symbol press **A**
- 0 This creates a symbol file in the directory of the schematic, navigate to this file and open it by writing (**xschem inverter.sym**)
- 0 Now you can drag the already existing lines to create the boundaries or use the tools seen in the toolbar to define custom shapes | 



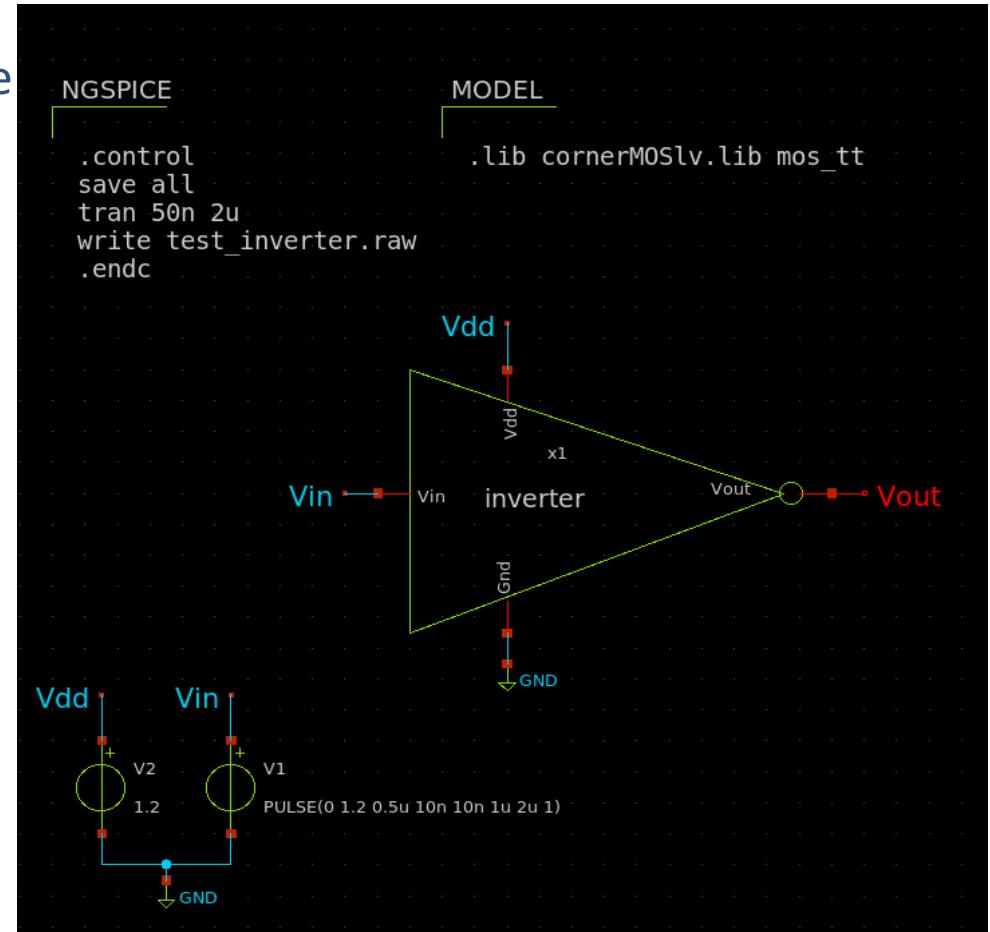
Inserting Symbol In Schematic



- 0 When the symbol is done remember to save and navigate over to the testbench file.
- 0 Now open component library and press **current dir**
- 0 Select your symbol and insert it in the testbench and connect accordingly and review the netlist. After this run the simulation!

```
.subckt inverter Vdd Vin Vout Gnd
*.iopin Vout
*.iopin Vdd
*.iopin Vin
*.iopin Gnd
XM1 Vout Vin Gnd Gnd sg13_lv_nmos w=1.0u l=0.45u ng=1 m=1
XM2 Vout Vin Vdd Vdd sg13_lv_pmos w=2.0u l=0.45u ng=1 m=1
...
```

- 0 Note that in the netlist the subcircuit definition is uncommented



Catching Up / Lunch Time !

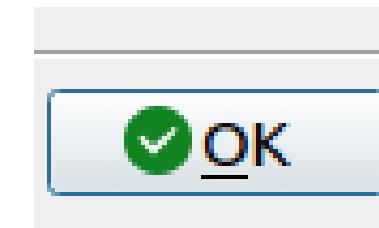
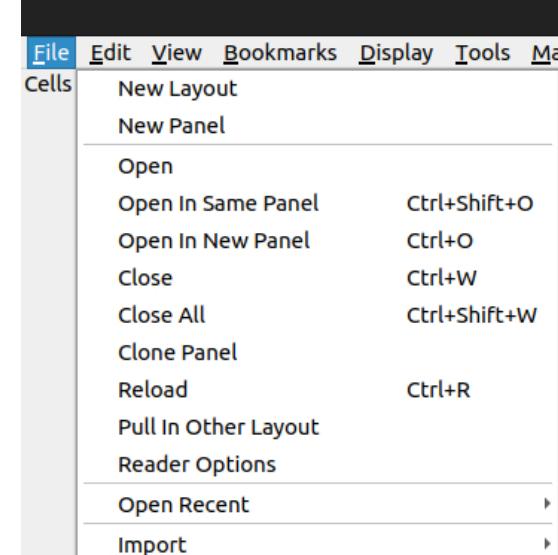
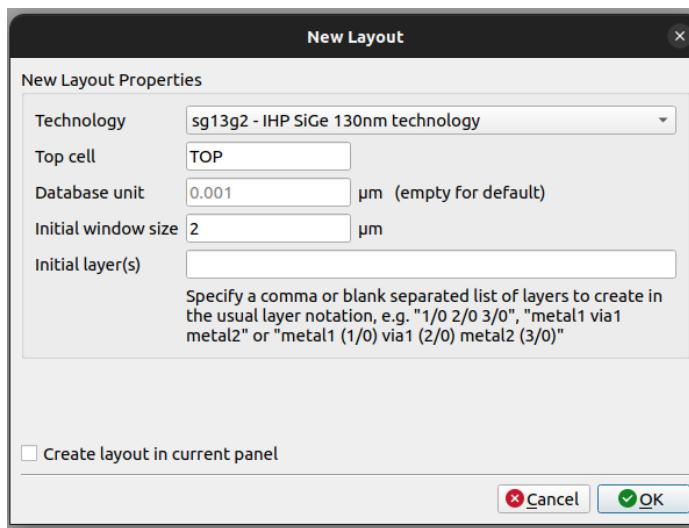
Next session:
Introduction to KLayout



Leibniz Institute
for high
performance
microelectronics

Getting Familiar With KLayout

Getting Started!

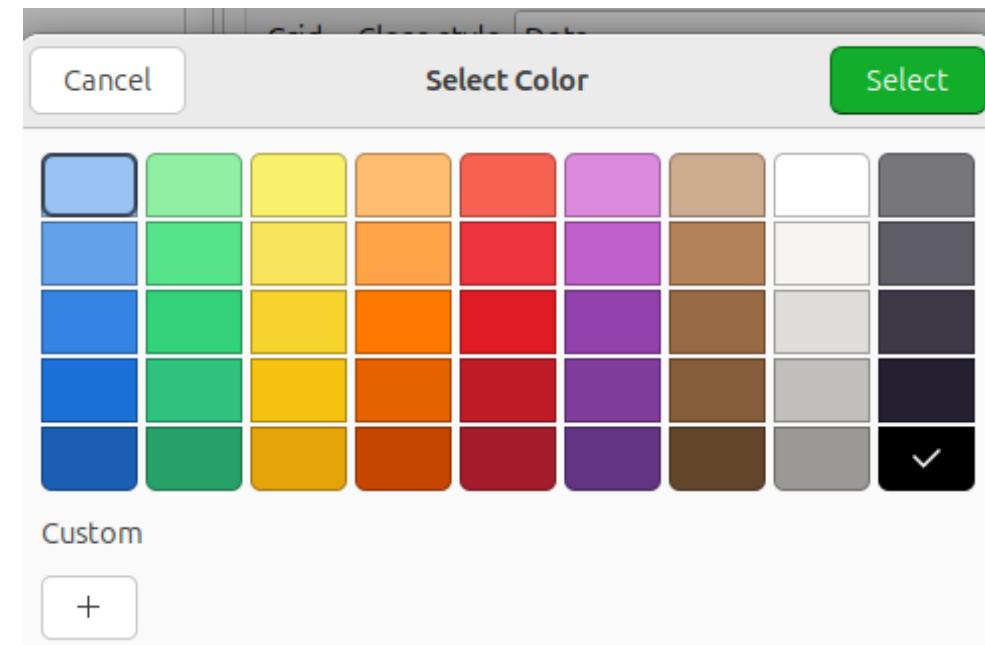
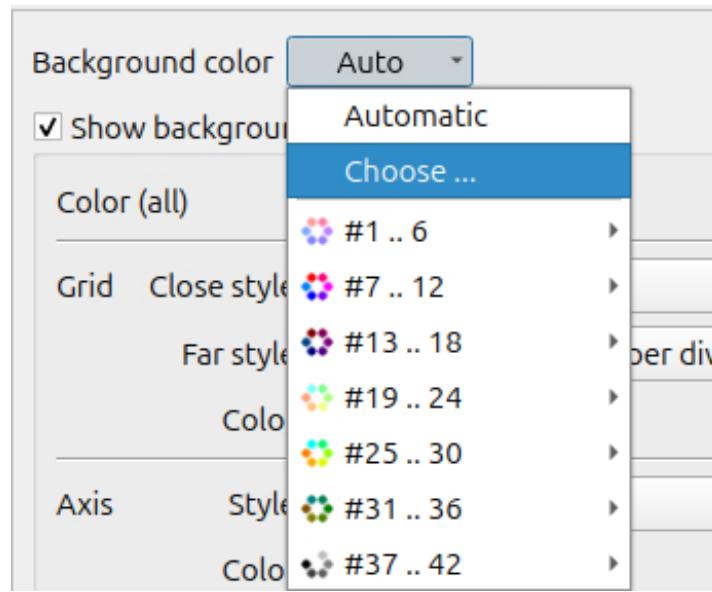


Launching Klayout: `klayout -e` (edit mode)

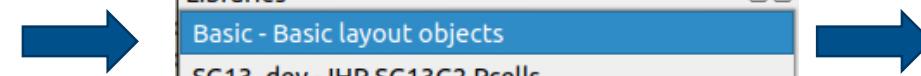
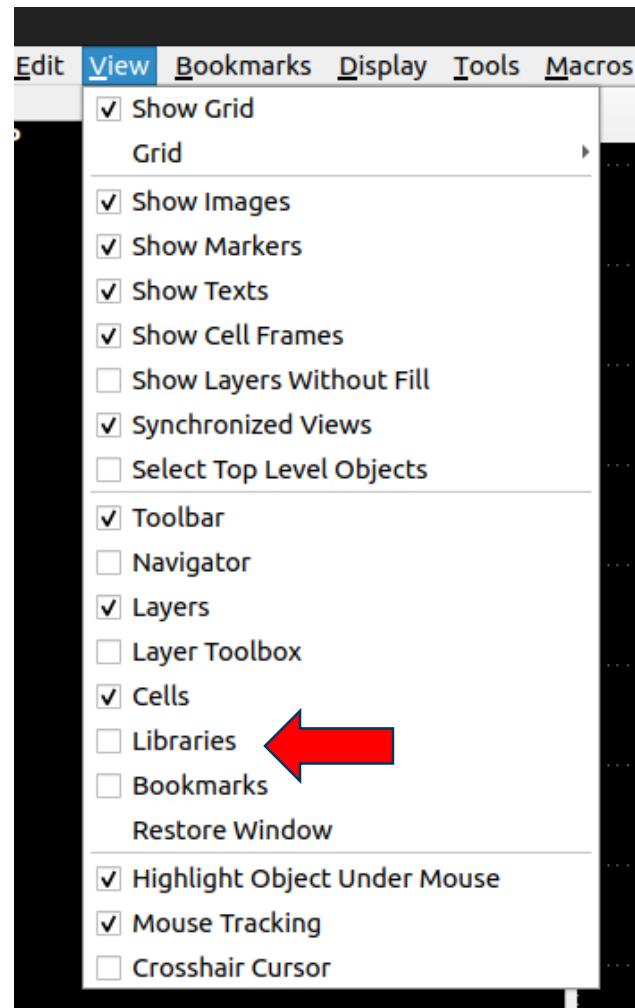
Pro Tip!



To set dark mode navigate to File -> Setup -> Display -> Background



Viewing The Devices



Drag and Drop!

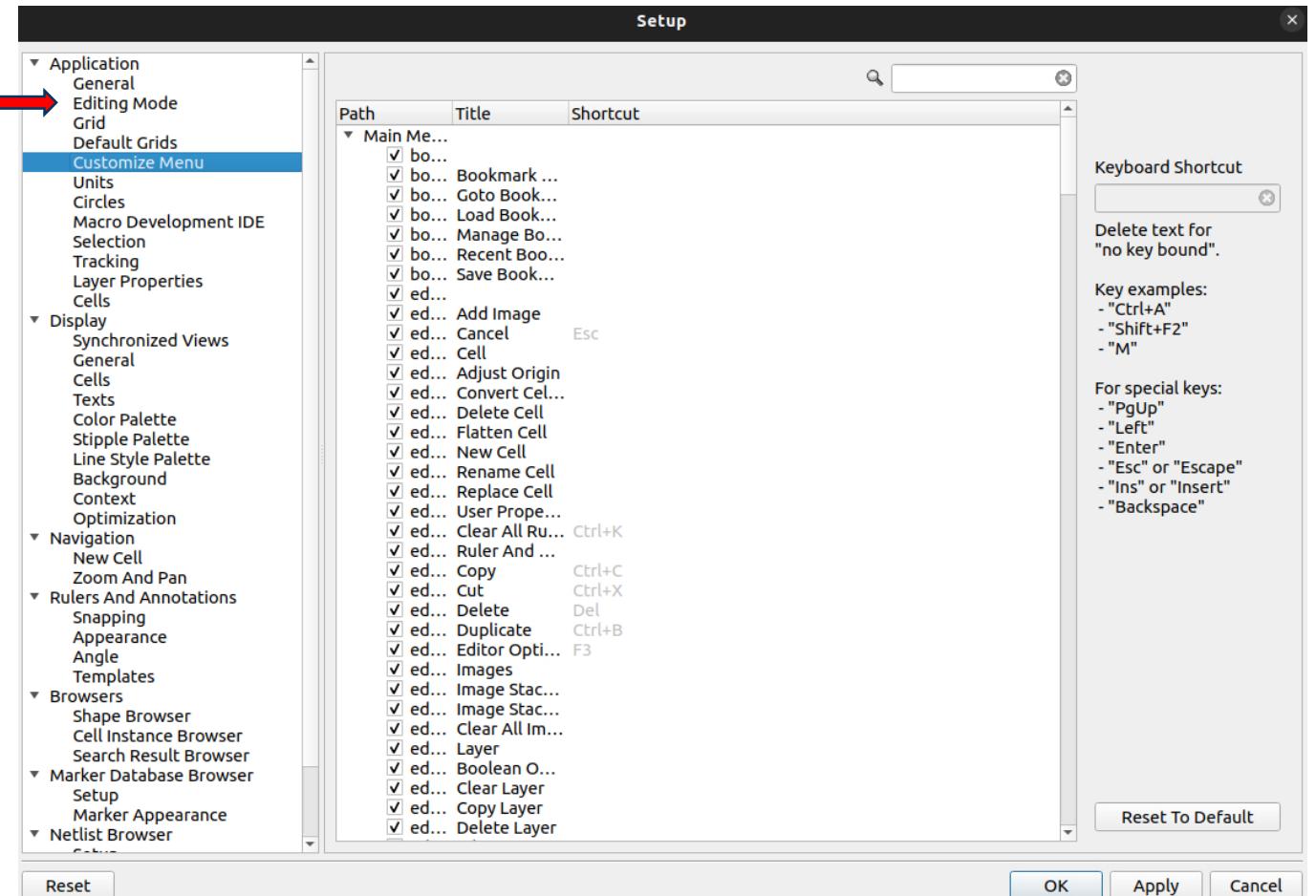


Want to set your own shortcuts?



Possible to set edit mode as
default

File -> Setup -> Customize Menu



Want to set your own shortcuts?



File -> Setup -> Customize Menu

The screenshot shows the 'Setup' dialog box with the 'Customize Menu' tab selected. On the left is a tree view of menu categories, and the main area is a table listing menu items with their titles and keyboard shortcuts. A sidebar on the right provides examples and special key definitions.

Path	Title	Shortcut
Main Me...	bo...	
	bo... Bookmark ...	
	bo... Goto Book...	
	bo... Load Book...	
	bo... Manage Boo...	
	bo... Recent Boo...	
	bo... Save Book...	
	ed...	
	ed... Add Image	
	ed... Cancel	Esc
	ed... Cell	
	ed... Adjust Origin	
	ed... Convert Cel...	
	ed... Delete Cell	
	ed... Flatten Cell	
	ed... New Cell	
	ed... Rename Cell	
	ed... Replace Cell	
	ed... User Propre...	
	ed... Clear All Ru...	Ctrl+K
	ed... Ruler And ...	
	ed... Copy	Ctrl+C
	ed... Cut	Ctrl+X
	ed... Delete	Del
	ed... Duplicate	Ctrl+B
	ed... Editor Opti...	F3
	ed... Images	
	ed... Image Stac...	
	ed... Image Stac...	
	ed... Clear All Im...	
	ed... Layer	
	ed... Boolean O...	
	ed... Clear Layer	
	ed... Copy Layer	
	ed... Delete Layer	

Keyboard Shortcut
Delete text for "no key bound".
Key examples:
- "Ctrl+A"
- "Shift+F2"
- "M"
For special keys:
- "PgUp"
- "Left"
- "Enter"
- "Esc" or "Escape"
- "Ins" or "Insert"
- "Backspace"

Reset To Default

OK Apply Cancel

Reset

Want to set your own shortcuts?



If shortcuts shown on next slide is not available from default you can set them yourself using the following example below

A screenshot of a software application window titled 'Keyboard Shortcut'. The main area is a table with columns 'Path', 'Title', and 'Shortcut'. A search bar at the top right contains the text 'hide'. The table shows two entries under 'Layer Pa...': 'hi... Hide All' and 'hi... Hide Empty... H'. The second entry is highlighted with a blue selection bar. To the right of the table is a panel titled 'Keyboard Shortcut' with a text input field containing 'H'. Below it is a note: 'Delete text for "no key bound".' Further down are sections 'Key examples:' and 'For special keys:', each listing several key combinations.

Path	Title	Shortcut
Layer Pa...	hi... Hide All	
Layer Pa...	hi... Hide Empty... H	

Keyboard Shortcut

H

Delete text for "no key bound".

Key examples:

- "Ctrl+A"
- "Shift+F2"
- "M"

For special keys:

- "PgUp"
- "Left"
- "Enter"
- "Esc" or "Escape"
- "Ins" or "Insert"
- "Backspace"



Simple maneuvering

- 0 **Launching Klayout:** klayout -e (edit mode)
- 0 **Zoom Fit (Main Menu):** F
- 0 **Move (Main Menu):** M
- 0 **Show Only Selected (layer panel):** Shift + Tab
- 0 **Show all used layers (layer panel):** Tab
- 0 **Hide Empty Layers (layer panel):** H
- 0 **Box (Main Menu):** W (click desired layer first)
- 0 **Partial (Main Menu):** P
- 0 **Ruler (Main Menu):** K
- 0 **Clear All Rulers And Annotations (Main Menu):** Ctrl + k

Simple maneuvering



Setup

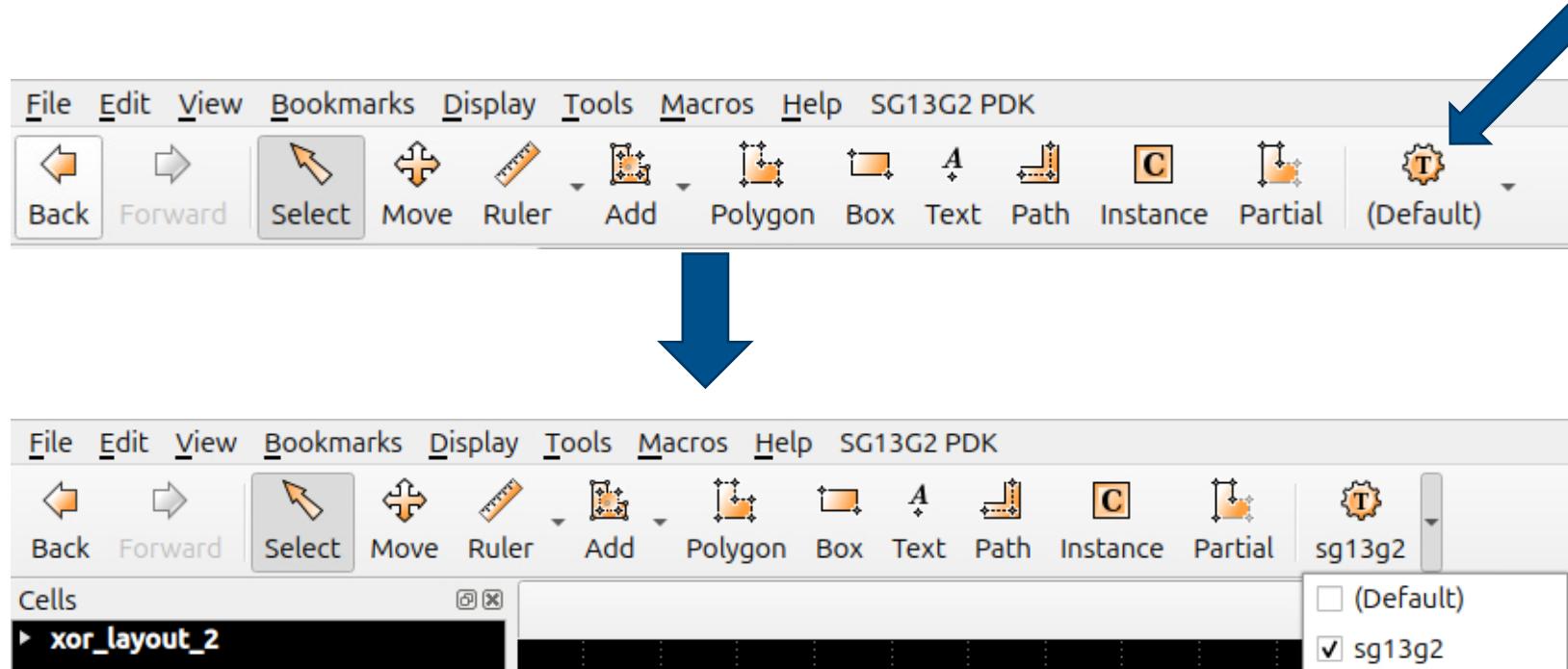
The screenshot shows the 'Setup' window with a sidebar containing various configuration categories. The 'Angle' category is currently selected, indicated by a blue bar at the bottom of the sidebar. The main panel displays 'Angle constraint (unless disabled in template)' with five radio button options: 'Any angle' (unselected), 'Orthogonal' (selected), 'Diagonal' (unselected), 'Horizontal only' (unselected), and 'Vertical only' (unselected). The sidebar categories include: Cells, Display (with Synchronized Views, General, Cells, Texts, Color Palette, Stipple Palette, Line Style Palette, Background, Context, Optimization), Navigation (with New Cell, Zoom And Pan), Rulers And Annotations (with Snapping, Appearance, Angle), Templates, and Browsers.

- Cells
- Display
 - Synchronized Views
 - General
 - Cells
 - Texts
 - Color Palette
 - Stipple Palette
 - Line Style Palette
 - Background
 - Context
 - Optimization
- Navigation
 - New Cell
 - Zoom And Pan
- Rulers And Annotations
 - Snapping
 - Appearance
 - Angle
- Templates
- Browsers

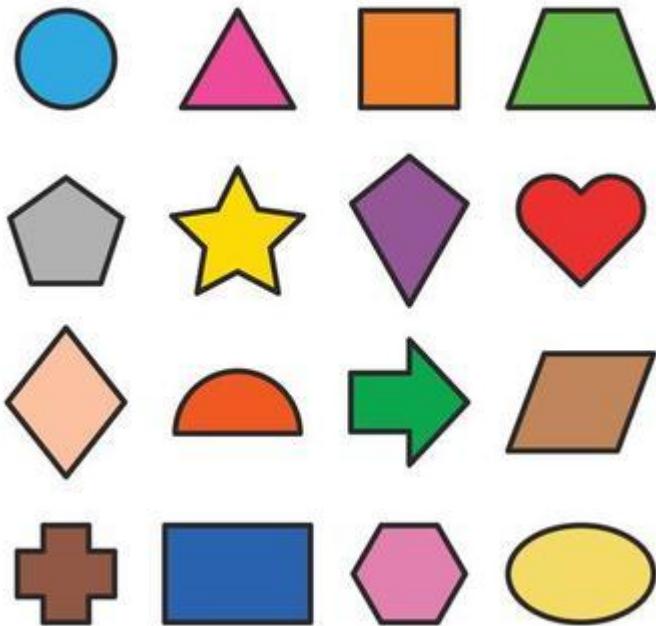
Angle constraint (unless disabled in template)

Any angle Orthogonal Horizontal only
 Diagonal Vertical only

Simple maneuvering



15-20 min playtime: Draw Cool Shapes/View Examples



Name	Last Commit Message
..	
FMD_QNC_AC3E_USM_TDBUCK.g...	upload gds files
FMD_QNC_Arlet6502.gds.gz	upload gds files
FMD_QNC_CryoChip.gds.gz	upload gds files
FMD_QNC_MARTIn.gds.gz	upload gds files
FMD_QNC_MS_Accelerator_AMux...	upload gds files
FMD_QNC_OTA_TIA.gds.gz	upload gds files
FMD_QNC_ROTest24_1.gds.gz	upload gds files
FMD_QNC_SG13G2Top.gds.gz	upload gds files
FMD_QNC_SingleCell_TestStructur...	upload gds files
FMD_QNC_merged_aes.filled.gds.gz	upload gds files
FMD_QNC_mmW_detector.gds.gz	upload gds files
README.md	
chip_r.gds	
chip_r_sealed.gds	
sealring.gds	
test_layout.gds	
.gitignore	
LICENSE	
README.md	Update README.md
chip_r.gds	
chip_r_sealed.gds	
sealring.gds	
test_layout.gds	

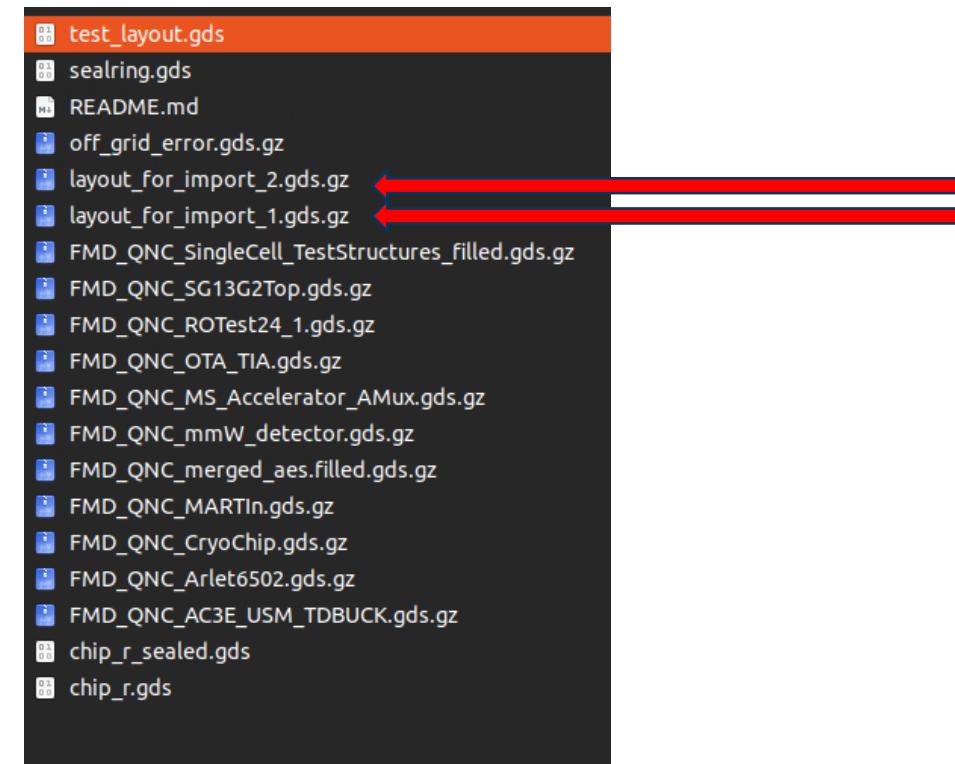
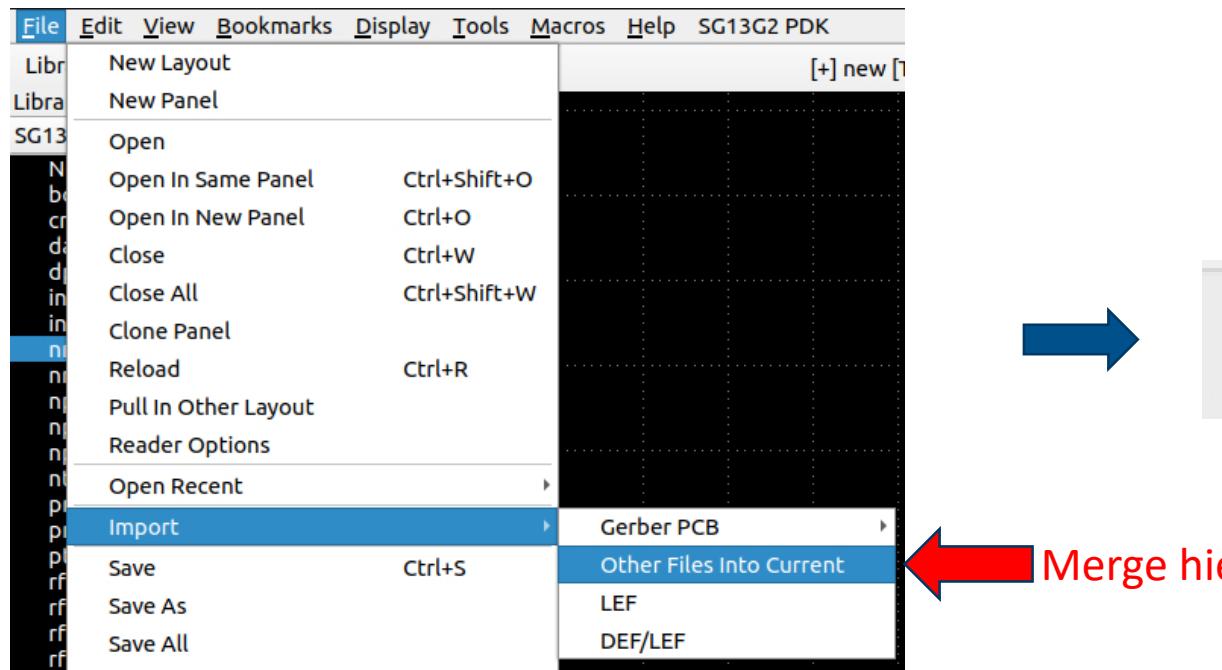
IHP-AnalogAcademy/utils

Example from command line: Klayout -e chip_r.gds

Alternatively: File -> Open -> chip_r.gds

Importing .gds files into current project

-0 At this point I need you to open a blank project and import the layouts into the project in the following way



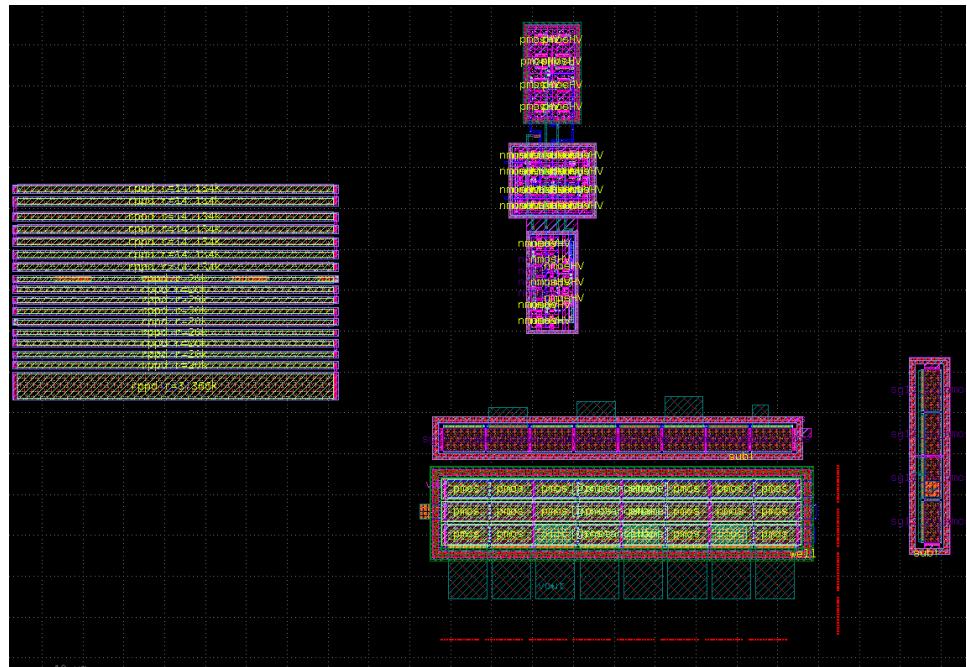
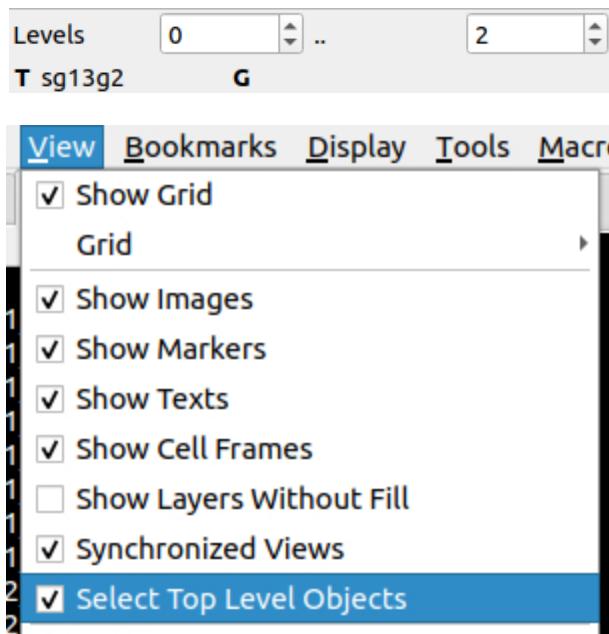
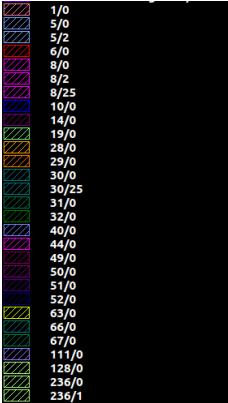
IHP-AnalogAcademy/utils

Small Overview Of The Chaos



- 0 At this point, the instances may be spaced apart due to different grids in each GDS. Press **F** to fit the view, then move the instances closer together by selecting instances, pressing **M**, and dragging from any marked instance.
 - 0 Remember to set the **Levels**: 

Ignore or delete the extra layers that was generated during the import

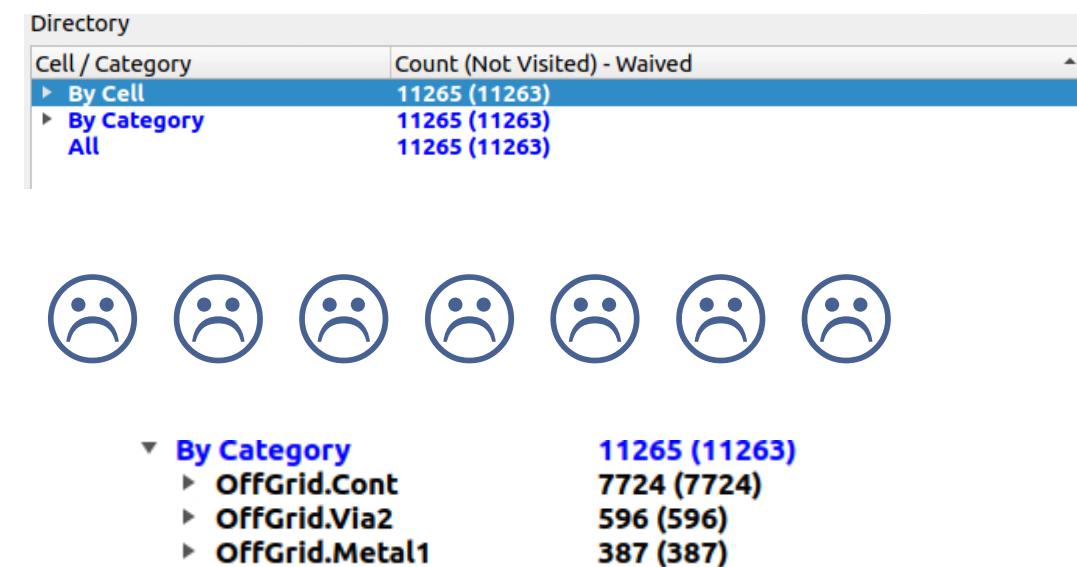


Running DRC ☺/☹



- There are two available scripts for running DRC: Max and Min. The minimum DRC deck (Critical design rules), ensures the absolute minimum requirements for fabrication. It's recommended to use the Max script for a DRC-clean design, as it covers a more thorough check

The screenshot shows the IHP Open PDK software interface. The top navigation bar includes Tools, Macros, Help, and SG13G2 PDK. The DRC menu is open, showing options like New DRC Script, Edit DRC Script, and a list of existing DRC scripts. One script, "/home/pedersen/IHP-Open-PDK/ihp-sg13g2/libs.tech/klayout/tech/drc/sg13g2_maximal.lydrc", is highlighted with a blue selection bar.



Overcoming Off Grid Errors



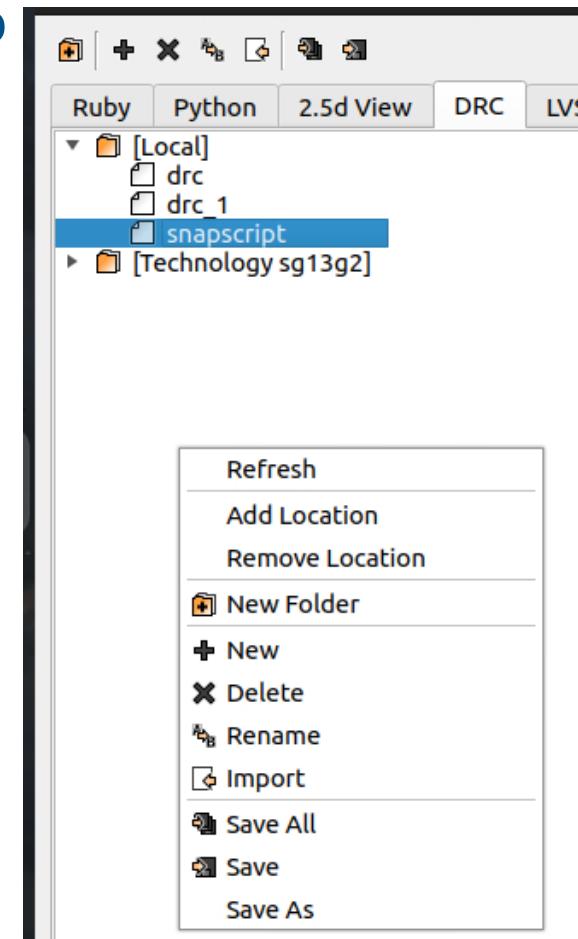
- 0 **Grid Issues in KLayout:** Importing GDS files into a project can cause grid errors
- 0 **Custom Grid in KLayout:** Every layout has a local grid in reference to the individual GDS
- 0 **Virtuoso to KLayout Export:** Similar grid adjustment is needed when exporting layouts around in commercial tools

What is the solution?

Overcoming Off Grid Errors



- 0 We must execute a ruby script that snaps selected instances on layers to the local grid
- 0 Navigate to macro development: F5
- 0 Right click the navigation pane and click: **import**
- 0 Now navigate to the *snapscript* folder (under **utils** folder from main directory) and import the *.lydrc* file



Overcoming Off Grid Errors



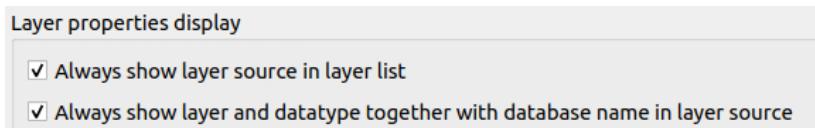
Layer Inputs

```
1 # Layer Inputs
2 # Activ = source.input("1/0")
3 # Activ_pin = source.input("1/2")
4 # Activ_mask = source.input("1/20") # Commented out as not on the list
5 # Activ_filler = source.input("1/22") # Commented out as not on the list
6 # Activ_nofill = source.input("1/23") # Commented out as not on the list
7 # BiWind = source.input("3/0") # Commented out as not on the list
8 # GatPoly = source.input("5/0")
9 # GatPoly_pin = source.input("5/2")
10 # GatPoly_filler = source.input("5/22") # Commented out as not on the list
11 # GatPoly_nofill = source.input("5/23") # Commented out as not on the list
12 Cont = source.input("6/0")
13 # NSD = source.input("7/0")
14 # NSD_block = source.input("7/21") # Commented out as not on the list
15 # Metall1 = source.input("8/0")
16 # Metall1_pin = source.input("8/2")
17 # Metall1_filler = source.input("8/22") # Commented out as not on the list
18 # Metall1_nofill = source.input("8/23") # Commented out as not on the list
19 # Metall1_slit = source.input("8/24") # Commented out as not on the list
20 # Passiv = source.input("9/0") # Commented out as not on the list
21 # Metal2 = source.input("10/0")
22 # Metal2_pin = source.input("10/2") # Commented out as not on the list
23 # Metal2_filler = source.input("10/22") # Commented out as not on the list
24 # Metal2_nofill = source.input("10/23") # Commented out as not on the list
```

Snapping Layers to Grid

```
112
113 #TopMetall1_pin.snap(5).output("126/2")
114 #TopMetall2.snap(5).output("134/0")
115 #TopMetall2_filler.snap(5).output("134/22")
116 #TopVia2.snap(5).output("133/0")
117 #Via1.snap(5).output("19/0")
118 #Via2.snap(5).output("29/0")
119 #Via3.snap(5).output("49/0")
120 #Via4.snap(5).output("66/0")
121 #TopVia1.snap(5).output("125/0")
122 #Vmim.snap(5).output("129/0")
123 #MIM.snap(5).output("36/0")
124 #Activ.snap(5).output("1/0")
125 #Activ_filler.snap(5).output("1/22")
126 #Activ_pin.snap(5).output("1/2")
127 Cont.snap(5).output("6/0")
128 #nSD.snap(5).output("7/0")
129 #pSD.snap(5).output("14/0")
130 #NWell.snap(5).output("31/0")
131 #ThickGate0x.snap(5).output("44/0")
132 #nBuLay.snap(5).output("32/0")
133 #PolyRes.snap(5).output("128/0")
134 #SalBlock.snap(5).output("28/0")
135 #EXTBlock.snap(5).output("111/0")
136 #dfpad.snap(5).output("41/0")
```

To view the indexes of the layers go to: File -> Setup -> Application -> Layer Properties



Run Script:





Better??

Directory	
Cell / Category	Count (Not Visited) - Waived
► By Cell	11265 (11263)
► By Category	11265 (11263)
All	11265 (11263)

Before

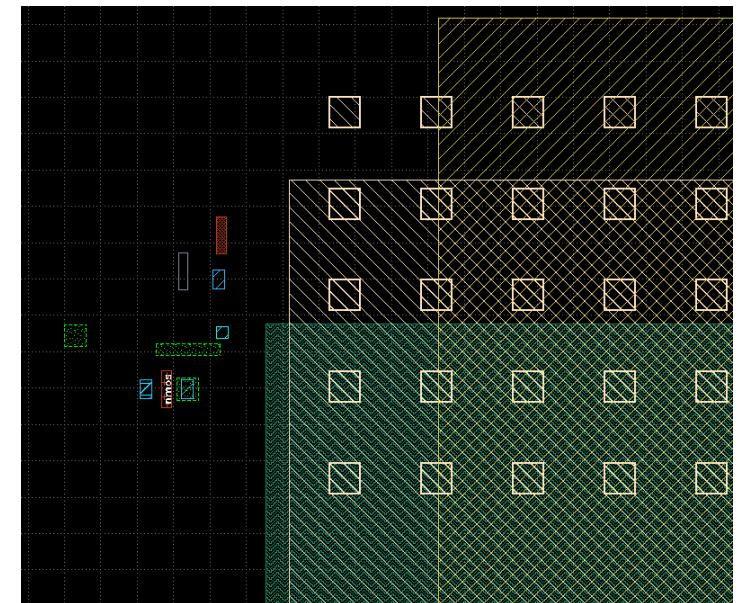
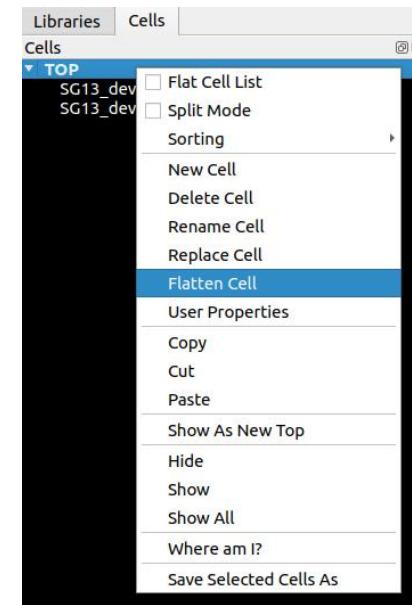
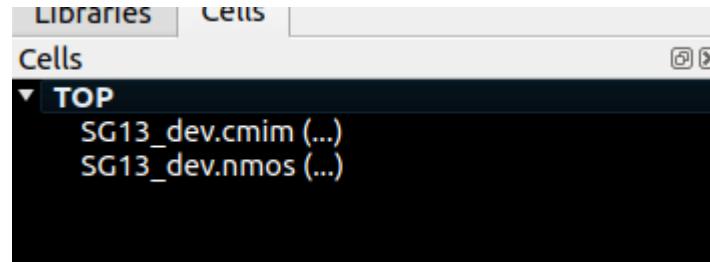
Cell / Category	Count (Not Visited) - Waived
► By Cell	3699 (3699)
▼ By Category	3699 (3699)

After

But why??

Flattening Your Design

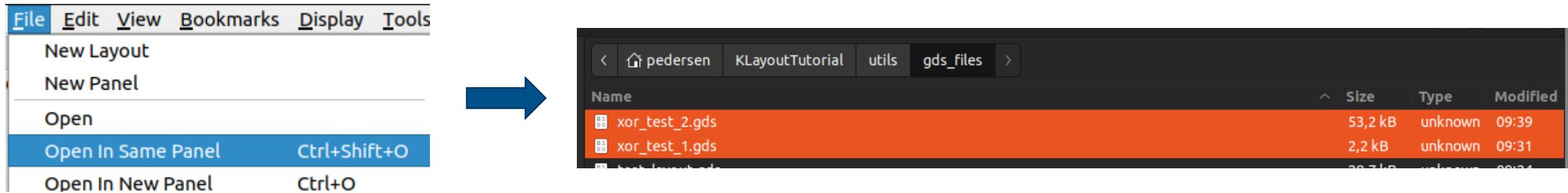
- 0 Combines all instances of sub-cells (child cells) into a single, flat structure, removing the hierarchical design. Why would we need this?
- 0 Prepares the design for export in formats that don't support this specific hierarchy, ensuring compatibility.



XOR Tool



- 0 **XOR Function in KLayout:** Performs a bitwise exclusive OR operation between two shapes, returning the area where the shapes differ.
- 0 **Use Cases:** Highlights differences between two layouts or masks. Helps identify discrepancies between design and manufacturing layers.



- 0 In this example its two almost identical designs!

XOR Tool



XOR Tool

Input

- Layout A: xor_test_1.gds[1], Cell 'layout_1'
- Layout B: xor_test_2.gds, Cell 'layout_2'
- Use for input: All Layers (different layouts)
- From region: All

Options

Compare modes:

- A XOR B (differences)
- A NOT B (In A but not in B)
- B NOT A (in B but not in A)
- Summarize missing layers (when output is marker DB)

Tolerances: Tolerances (t1,t2,..) in micron

Hierarchical: consider hierarchy (experimental)

Tiling: Tile size in micron Heal result shapes

Threads: 1 Used for tiles and layers

Output

Send output to: Marker database with offset

Tools -> XOR Tool

Marker Database Browser

Database: XOR (Comparison of 'xor_test_1.gds[1], Cell layout_1' vs. 'xor_test_2.gds, Cell layout_2')
... on layout: xor_test_1.gds[1]

Markers

Cell / Category	Count (Not Visited) - Waived
By Cell	1
By Category	1
XOR	1
All	1

Info

1/0 [layout_1]
Results for layer 1/0
polygon: (-1.035,3.895;-1.035,6.58;1.445,6.58;1.445,3.895)

Try with two identical cells to see 0 errors 😊



Settings Up the LVS (Live Demo)

- 0 Open Xschem and create a schematic of your own choice
- 0 Navigate to the Simulation tab and under LVS select the following →
- 0 Then press *Set top level netlist name* and name it with .cdl extension
- 0 Now press netlist. This creates the cdl file and places it in the simulation folder
- 0 Now launch Klayout and navigate to SG13G2 PDK -> LVS options
- 0 Browse Netlist path and point to the .cdl file(make sure top cell name is the same as in netlist)
- 0 Now you run the LVS!

✓ LVS netlist + Top level is a .subckt
✓ Upper case .SUBCKT and .ENDS
Top level is a .subckt
Set 'lvs_ignore' variable
Use 'spiceprefix' attribute

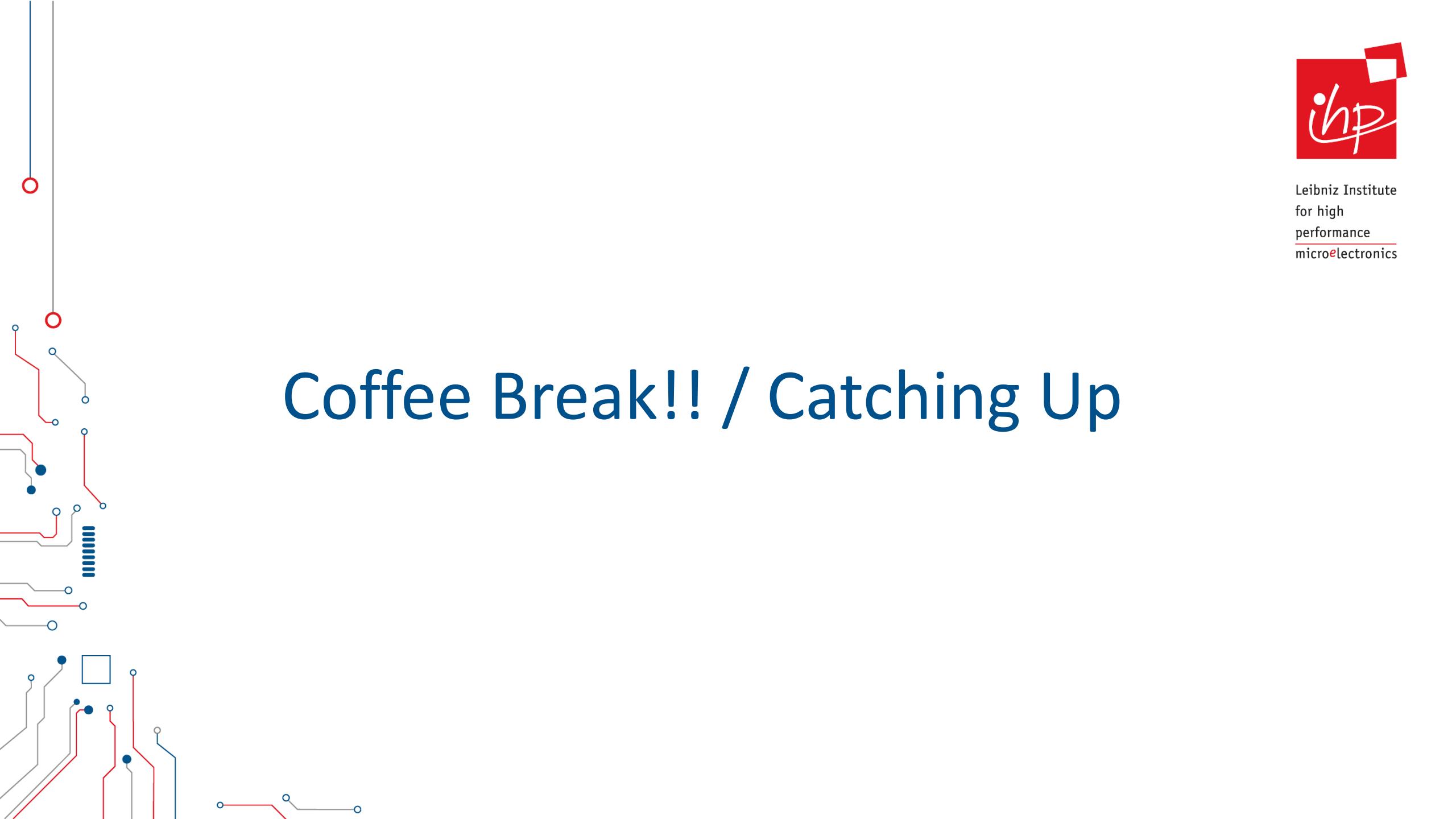
Reference: utils/lvs_tester (provides LVS clean example)

Questions?





Leibniz Institute
for high
performance
microelectronics

A faint background diagram of a microelectronic circuit is visible, featuring various colored lines (red, blue, grey), small circles, and a central vertical stack of blue rectangles representing capacitors or resistors.

Coffee Break!! / Catching Up

What Will We Do Tomorrow?



- 0 **OTA Design & Analysis:** Perform DC and AC analysis to evaluate key design metrics.
- 0 **Bandgap Voltage Reference:** Conduct DC, transient, and mismatch analysis using Monte Carlo sampling.
- 0 **From Design to Tape-Out:** Explore the final layout and discuss the steps required for tape-out in the open-source domain.
- 0 **Layout Competition:** Compete to design the best OTA layout with the fewest DRC/LVS errors.

Hands On Session!!



- 0 Complete the Schematic (Inverter):** Finalize the inverter schematic and explore Xschem simulations. For reference, consult the Ngspice documentation:
ngspice.sourceforge.io/docs/ngspice-manual.pdf
- 0 From Simulation to Layout:** Transition from simulation to creating a basic layout of the inverter. (Assistance will be provided during the session). Physical verification is not a must at this stage
- 0 Prepare for Tomorrow's Session:** Optionally, begin working on tasks for tomorrow's session! Navigate to:
IHP-AnalogAcademy/modules/module_1_bandgap_reference/part_1 OTA/design_of_ota.md