



Leibniz Institute
for high
performance
microelectronics

Working with the IHP-Open-PDK with Open-Source Tools

Krzysztof Herman (IHP-Open-PDK developer)

CEITEC - 24/01/2025

Projects: BMBF → FMD-QNC (16ME0831)
VDE/VDI → IHP Open130-G2 (16ME0852)

Table of contents



Introduction to the IHP

Our motivation and goals in the open source silicon domain

Description of the open source solutions developed and supported by IHP

Understanding the power of the community

Planification of Open MPW runs

Lessons learned after three months and future activities

IHP in a nutshell



- IHP is the European research and innovation centre for silicon-based systems, ultrahigh-frequency circuits and technologies,
- 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.
- Qualified technological platform with direct access for science and industry
- Vertical structure from material research to system architecture
- 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

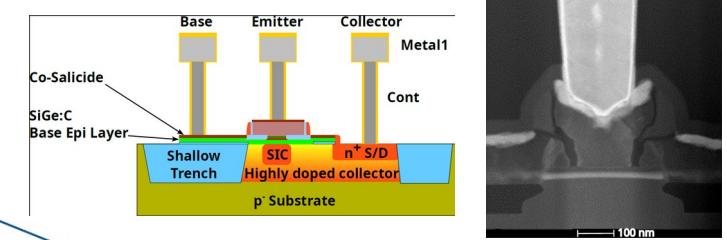




Table of contents

Introduction to the IHP

Our motivation and goals in the open source silicon domain

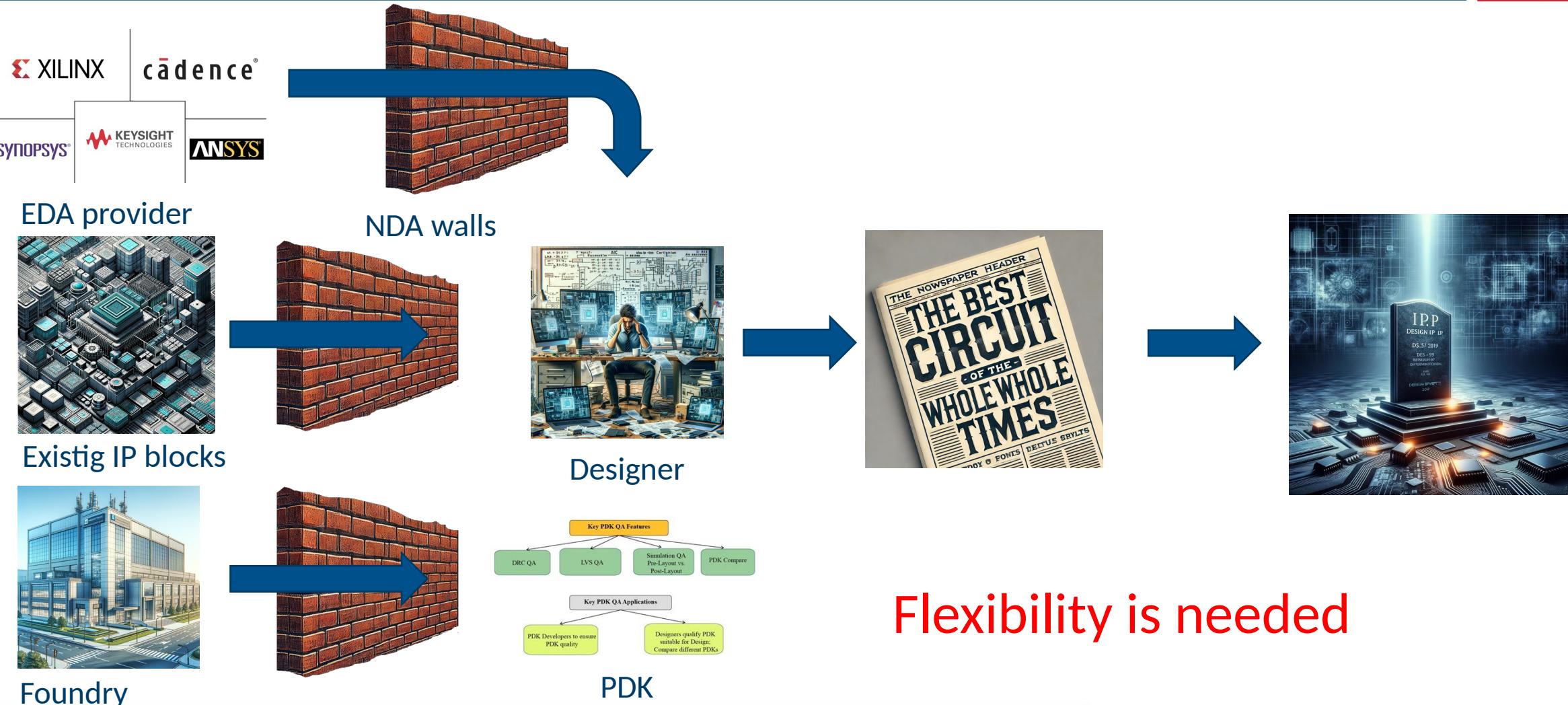
Description of the open source solutions developed and supported by IHP

Understanding the power of the community

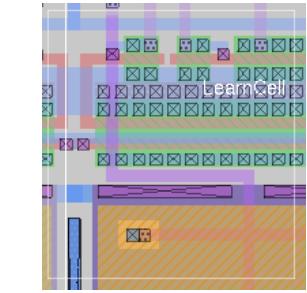
Planification of Open MPW runs

Lessons learned after three months and future activities

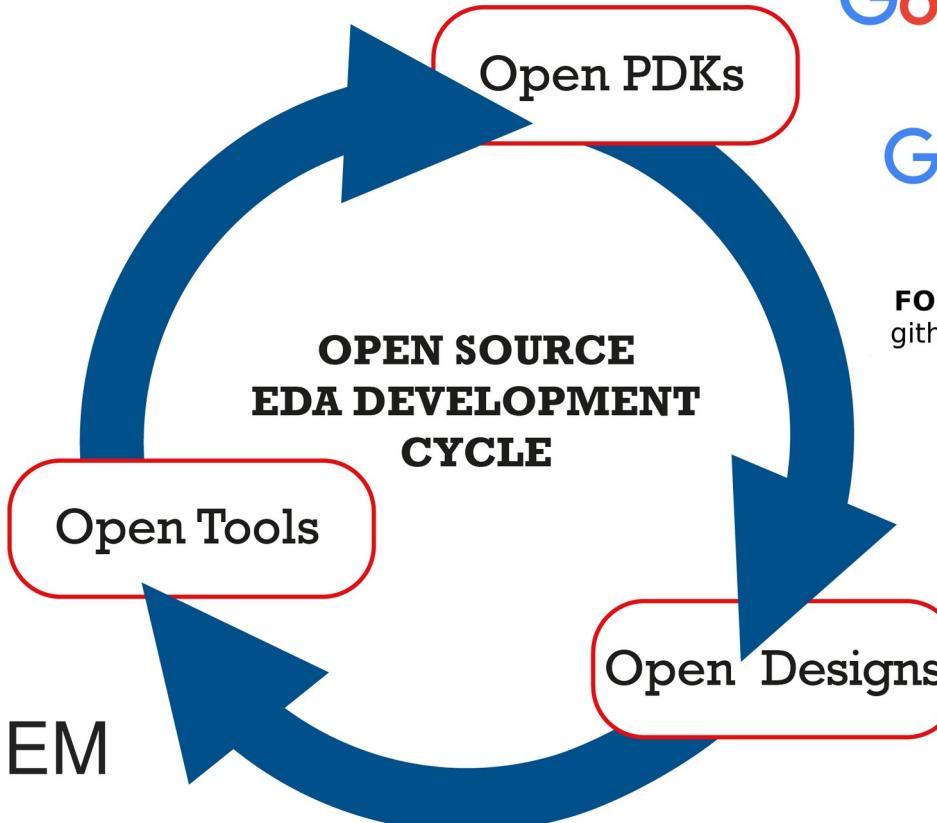
Traditional ASIC development (academic perspective)



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

Working the IHP Open PDK



Welcome to IHP 130nm BiCMOS Open Source PDK documentation!

⚠ Warning

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 libtool 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>

Getting started really isn't that hard!



```
pedersen@IHP-OPDK:~$ xschem
Sourcing /usr/local/share/xschem/xschemrc init file
Sourcing /home/pedersen/.xschem/xschemrc init file
SG13G2_MODELS: /home/pedersen/IHP-Open-PDK/ihp-sg13g2/libs.tech/ngspice/models
SG13G2_MODELS_XYCE: /home/pedersen/IHP-Open-PDK/ihp-sg13g2/libs.tech/xyce/models
xschem [~]
```

DC

- dc_lv_nmos x5
- dc_hv_nmos x6
- dc_lv_pmos x7
- dc_hv_pmos x8
- dc_mos_temp x11
- dc_mos_cs_temp x12
- dc_res_temp x13
- dc_ntap1 x25
- dc_ptap1 x26
- dc_diode_op x14
- dc_diode_temp x15
- dc_hbt_13g2 x17
- dc_pnpMPA x30
- dc_logic_not x28

Transient

- tran_mim_cap x10
- tran_logic_not x27
- tran_logic_nand x29

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
```

MODEL
.lib cornerMOSlv.lib mos_tt

i(vd)

load waves Ctrl + left click

```
dc_lv_nmos.spice" -a || sh
```

```
** Copyright 2001-2024. The ngspice team.
** Please get your ngspice manual. From https://ngspice.sourceforge.io/docs.html
** Creation Date: Tue Jul 18 16:09:25 2024 UTC 2024
****

Note: No compatibility mode selected!

Circuit: ** sch_path: /home/pedersen/IHP-Open-PDK/ihp-sg13g2/libs.tech/xschem/sg13g2_tests/dc_lv_nmos.sch
Doing analysis at TEMP = 27,000000 and TH0 = 27,000000
Using SPARSE 1.3 as Direct Linear Solver
No. of Data Rows : 1
Doing analysis at TEMP = 27,000000 and TH0 = 27,000000
Using SPARSE 1.3 as Direct Linear Solver
No. of Data Rows : 609
binary raw file "dc_lv_nmos.raw"
logplot dc_lv_nmos.raw
```

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



Table of contents



Introduction to the IHP

Our motivation and goals in the open source silicon domain

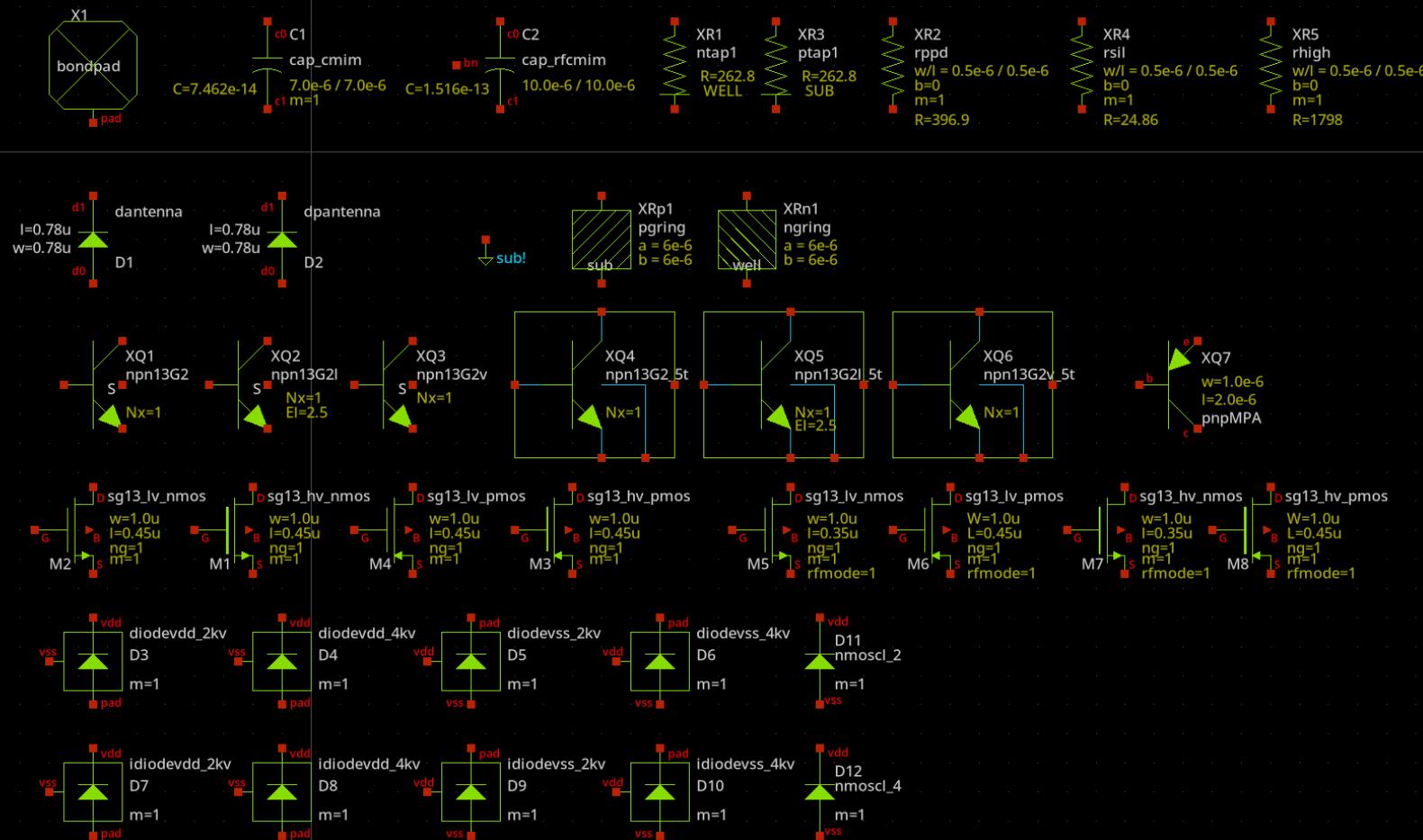
Description of the open source solutions developed and supported by IHP

Understanding the power of the community

Planification of Open MPW runs

Lessons learned after three months and future activities

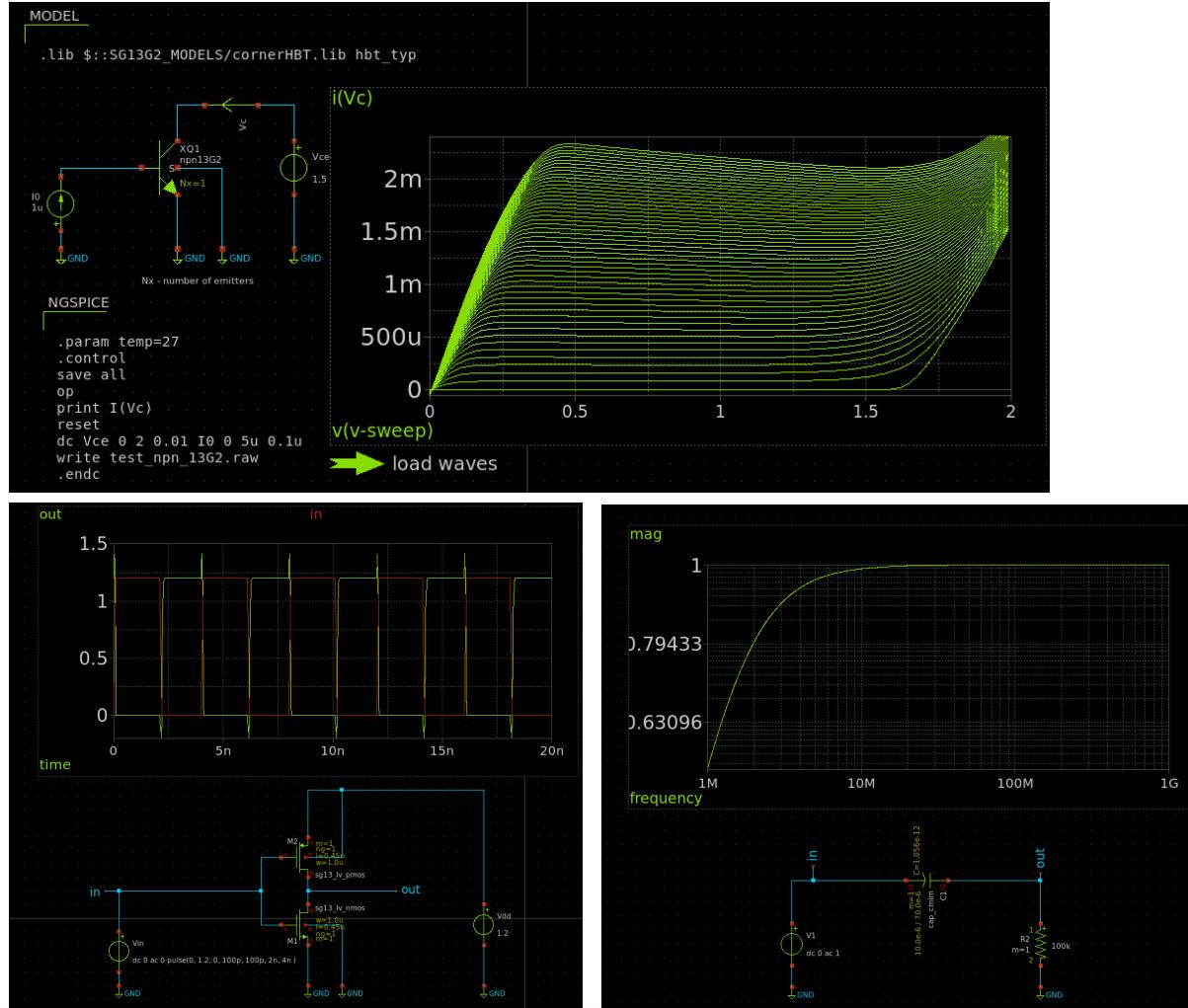
OpenPDK support for schematic capture



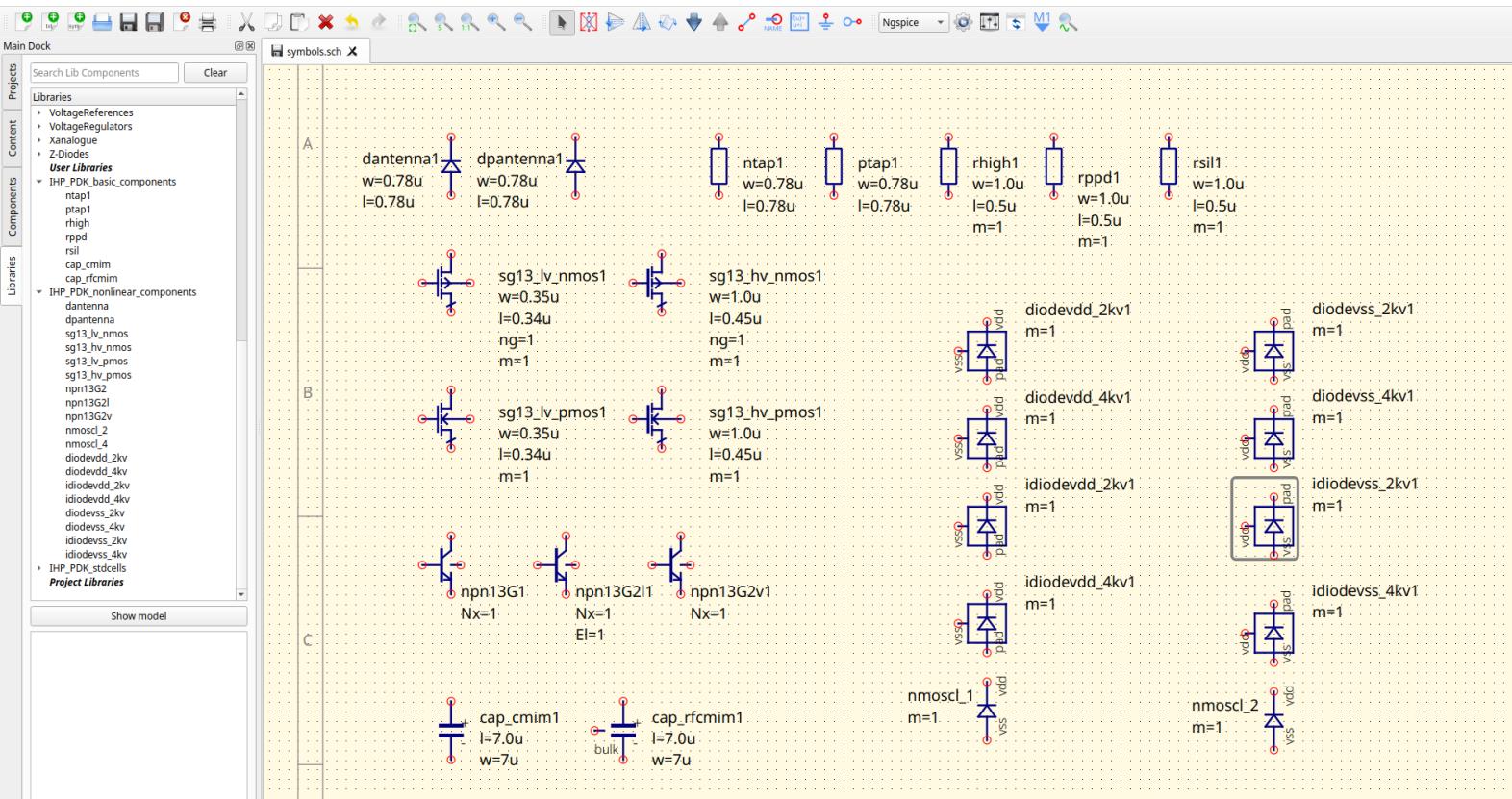
The current version of the IHP OpenPDK supports:

- 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



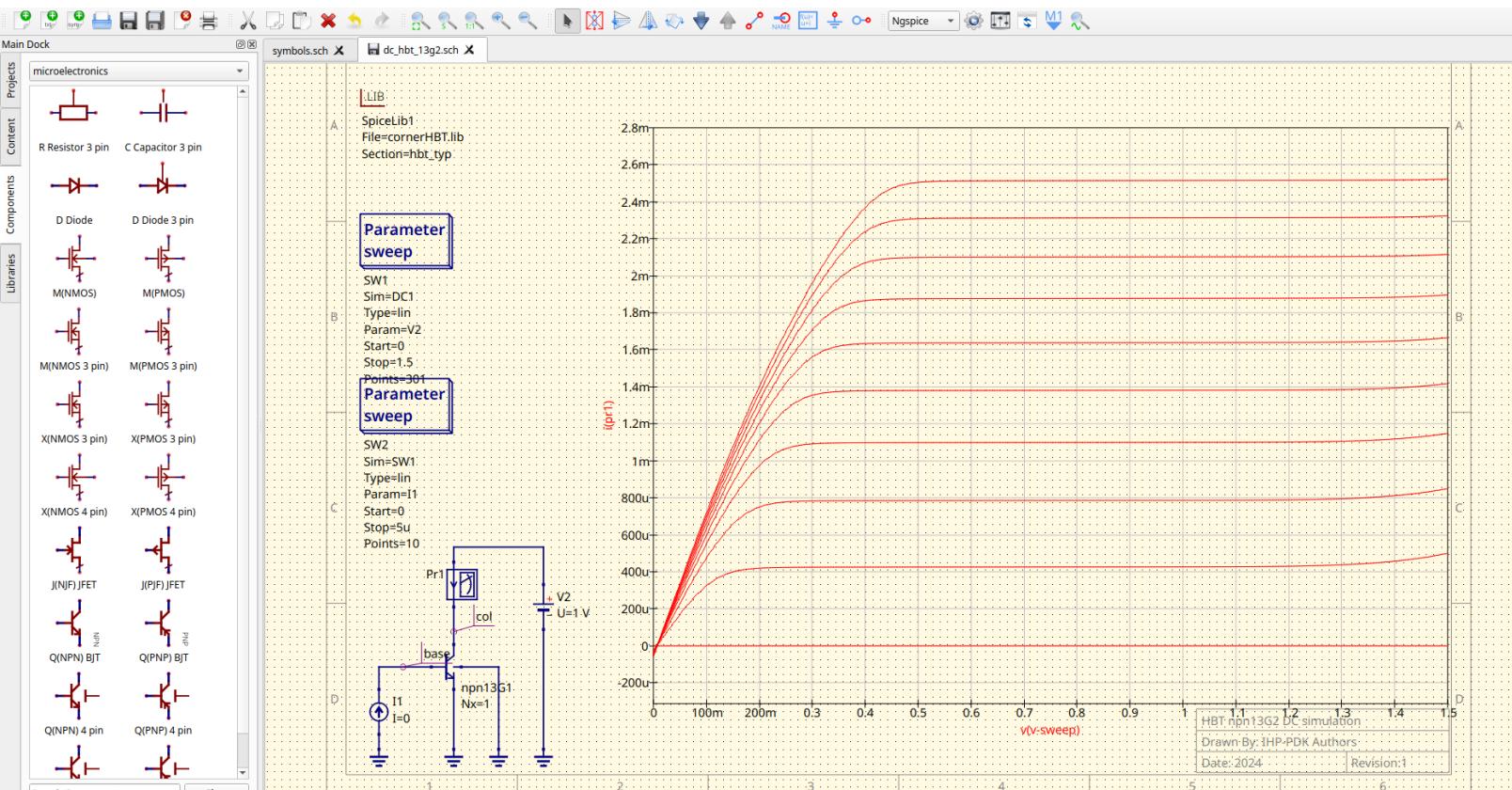
OpenPDK support for Qucs-S schematic capture



The current version of the IHP OpenPDK supports:

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

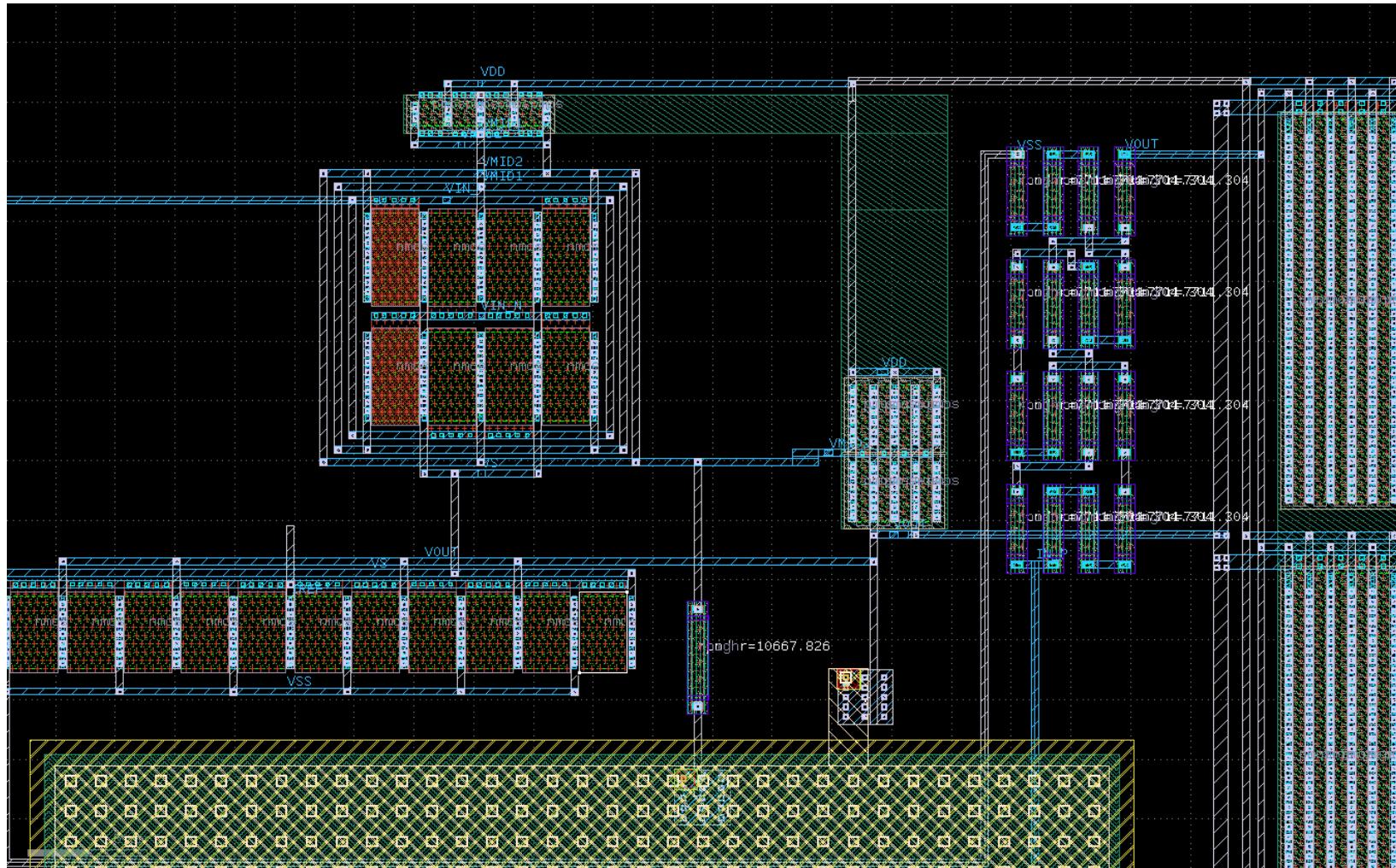
OpenPDK support for Qucs-S schematic capture



Qucs-S development:

- 0 Agnostic support for PDK
- 0 CDL netlisting
- 0 Interoperability with OpenEMS and Klayout
- 0 Verilog-A ...

KLayout – primary tool for open source layout design

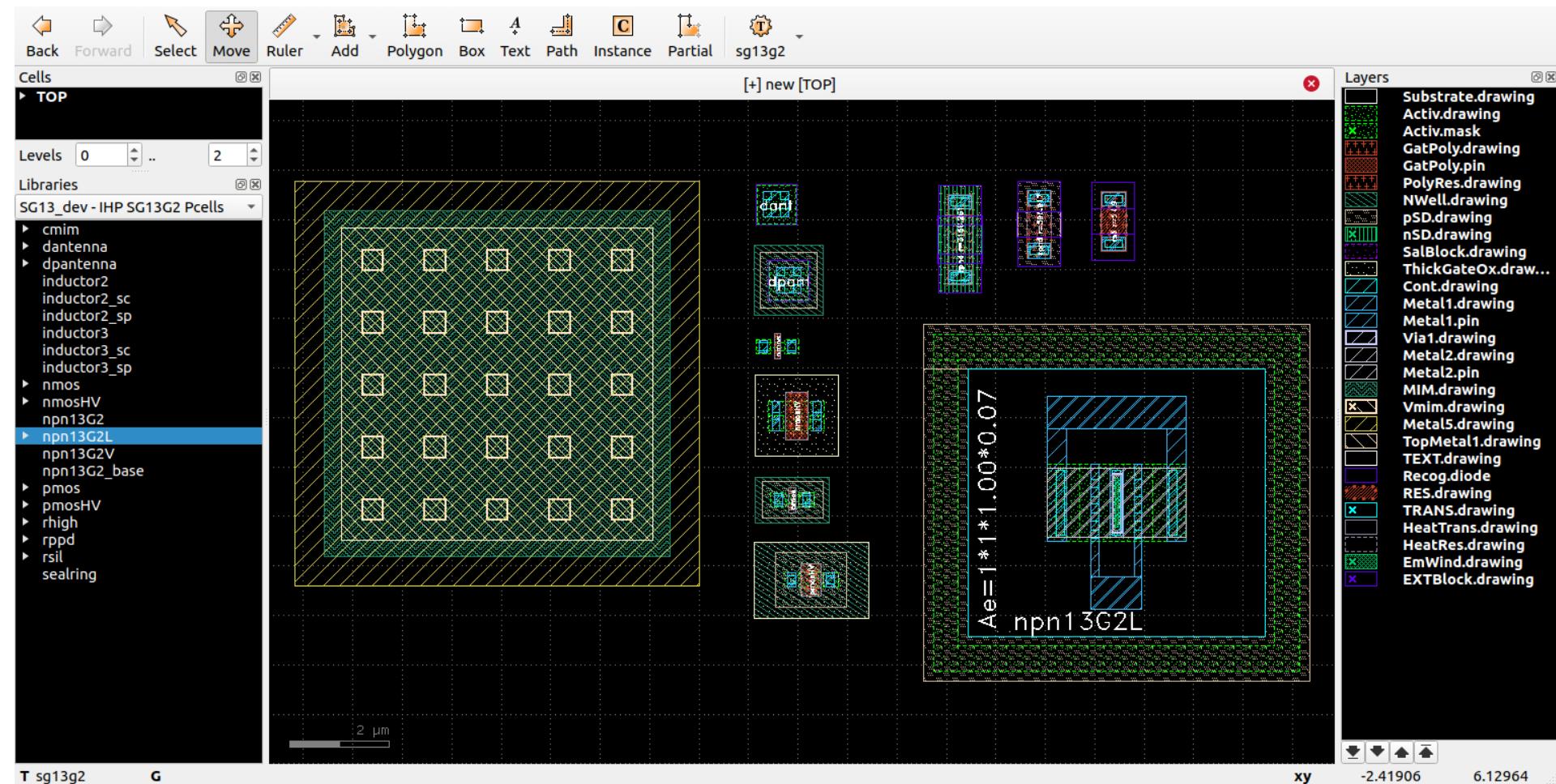


KLayout key features:

- 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



- o Pycells compatible with Synopsys PyCell Studio and Keysight ADS
- o 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]".

Left Panel (Main View): Displays the layout editor window with a grid background. A 1 μm scale bar is visible at the bottom left. The "Cells" panel shows the hierarchy: **gatpoly** (selected), metal1, nwell. The "Layers" panel lists the following layers:

- Activ.drw
- GatPoly.drw
- GatPoly.flr
- Cont.drw
- Metal1.drw
- Metal1.pin
- pSD.drw
- NWell.drw
- Substrate.drw
- ThickGateOx.drw
- TEXT.drw

The "Layer Toolbox" on the right side of the layout editor provides options for Color, Frame color, Stipple, Animation, Style, and Visibility.

Right Panel (Marker Database Browser): Shows the "Marker Database Browser" window with the following details:

- Database:** sg13g2.lyrdb
- ... on layout:** sg13g2_qacells.gds
- Directory:** Cell / Category
- Markers:** F I W
- Marker Summary:** 37 (35) total markers found.
- Marker Details:** Gat.d [gatpoly] has 8 (8) markers, including:
 - 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:** All categories have 37 (35) markers.

Bottom Right: A small red circular icon with a white dot is located in the bottom right corner.

KLayout example LVS run on sg13_lv_nmos mosfets



Netlist LVS

... on layout sg13_lv_nmos.gds

Netlist Schematic Cross Reference Log

Circuits sg13_lv_r

Objects	Layout	Reference
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]	
► -o S ⇌ D	\$35 (1)	D6 (2)
► -o D ⇌ S	\$36 (1)	S6 (2)
► -o G	\$37 (1)	G6 (2)
► -o 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]	

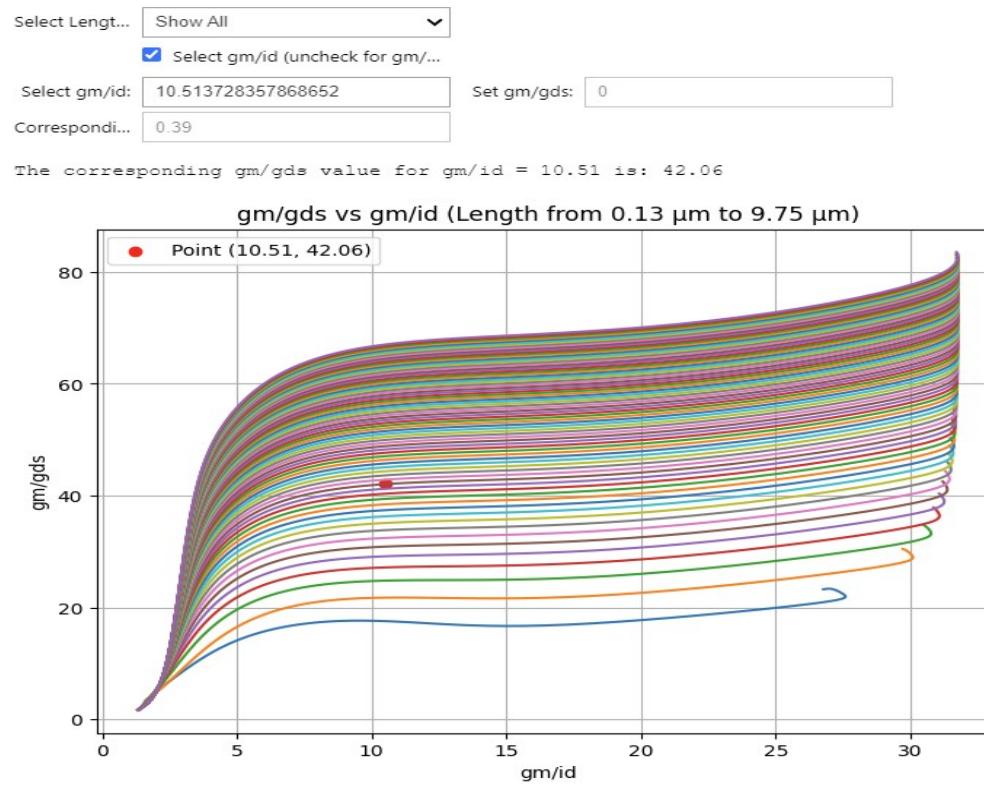
Klayout LVS ruledeck:

- accepts CDL netlist,
 - extracts devices from layout,
 - creates extracted netlist,
 - performs checks and compares:
 - Pins
 - Nets
 - Devices
 - reports inconsistencies
 - 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

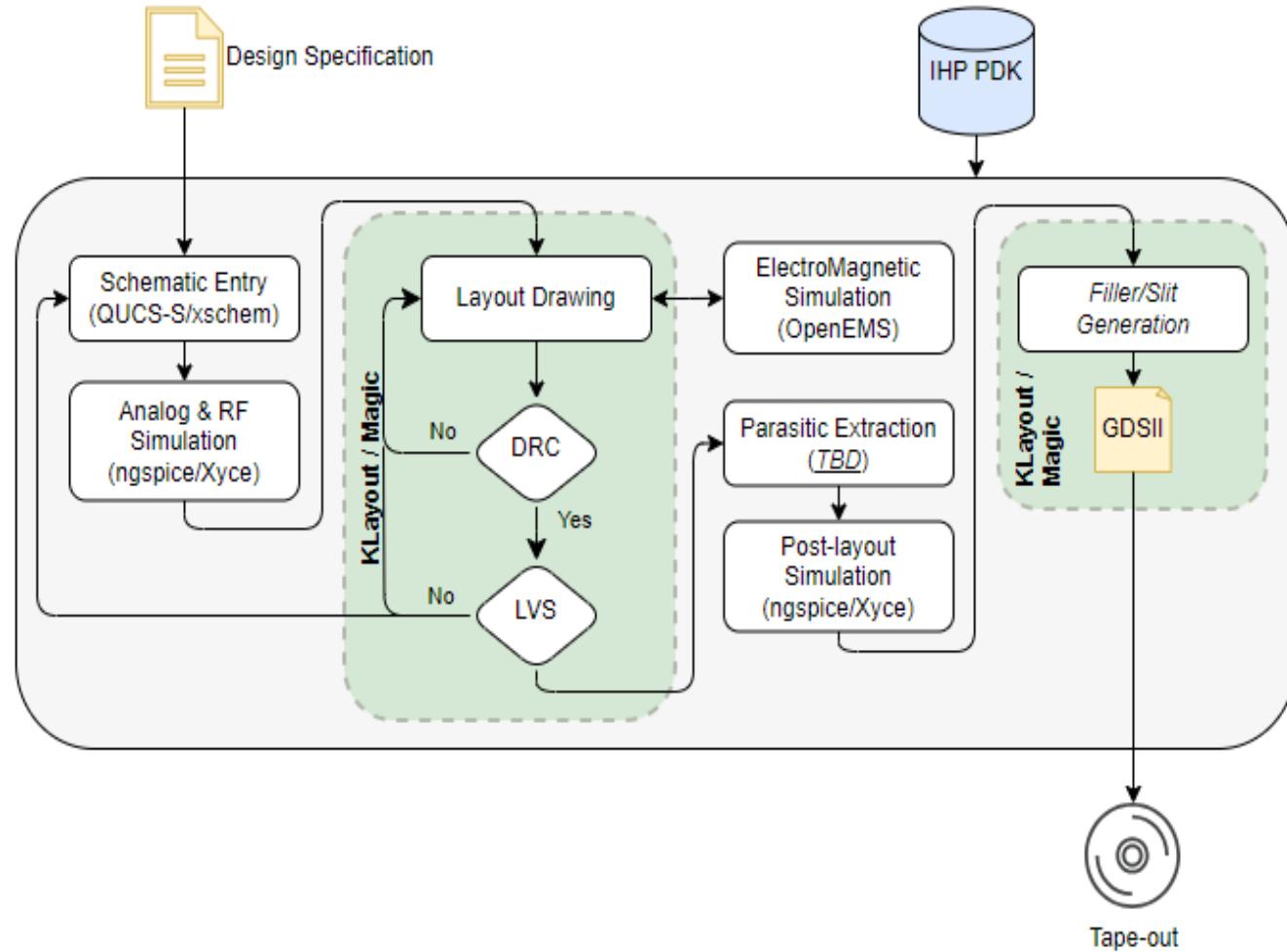
Layout Automation

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

What about the Tools for Radio Frequency Design?

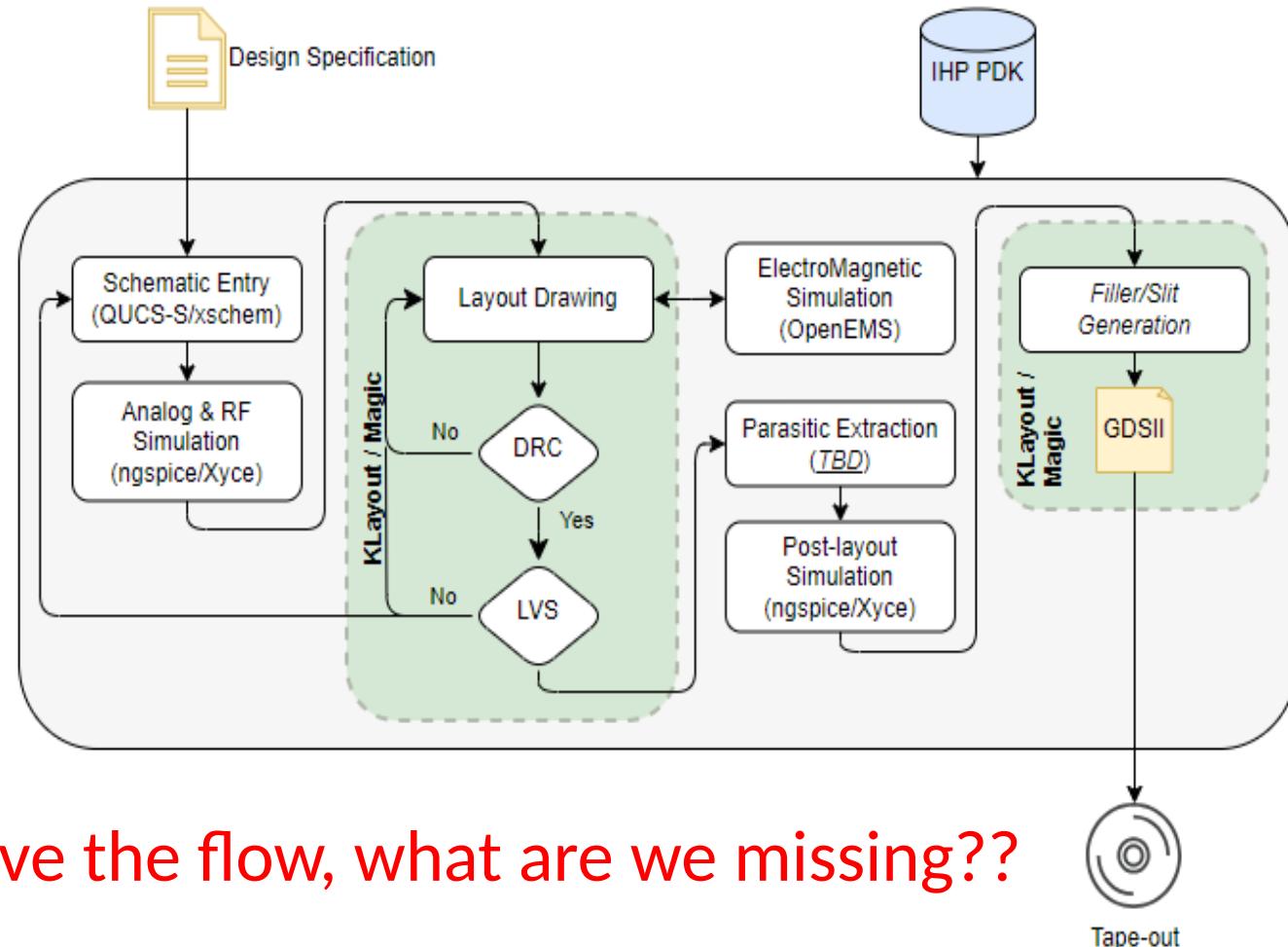


Analog/RF Open Source Design Flow: status at a glance



- 0 **KLayout-oriented flow**
 - 0 Layout design
 - 0 Parameterizable cells
 - 0 Physical Verification
- 0 QUCS-S, xschem
- 0 ngspice, Xyce
- 0 OpenEMS, Elmerfem?
- 0 Working on a faster solver for EMS (ELMER)

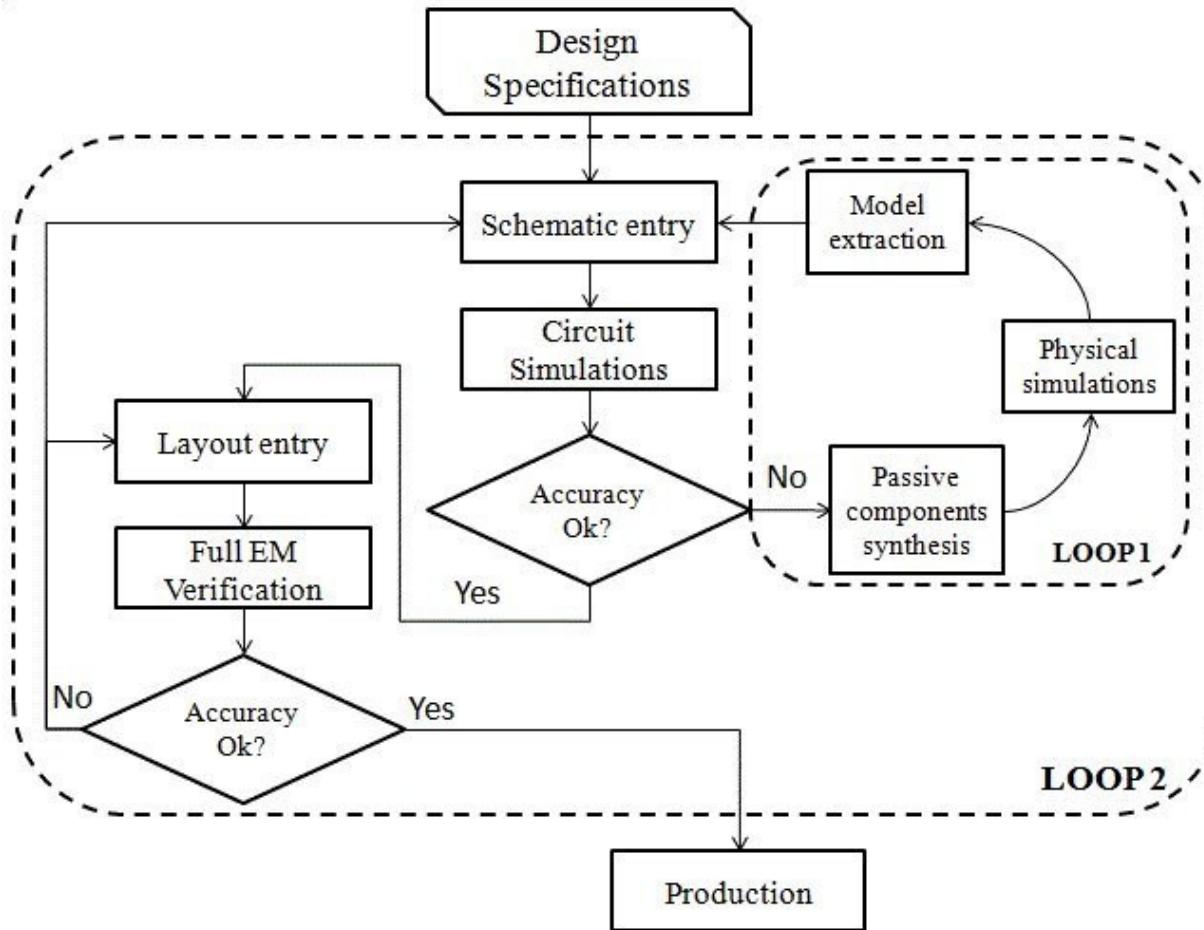
Analog/RF Open Source Design Flow: status at a glance



- 0 KLayout-oriented flow
 - 0 Layout design
 - 0 Parameterizable cells
 - 0 Physical Verification
- 0 QUCS-S, xschem
- 0 ngspice, Xyce
- 0 OpenEMS, Elmerfem?
- 0 Working on a faster solver for EMS (ELMER)

We have the flow, what are we missing??

Streamlining The Process of Design



- Repetition highlights the need for efficiency!
- We need integration of tools
- 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



Table of contents

Introduction to the IHP

Our motivation and goals in the open source silicon domain

Description of the open source soultions developed and supported by IHP

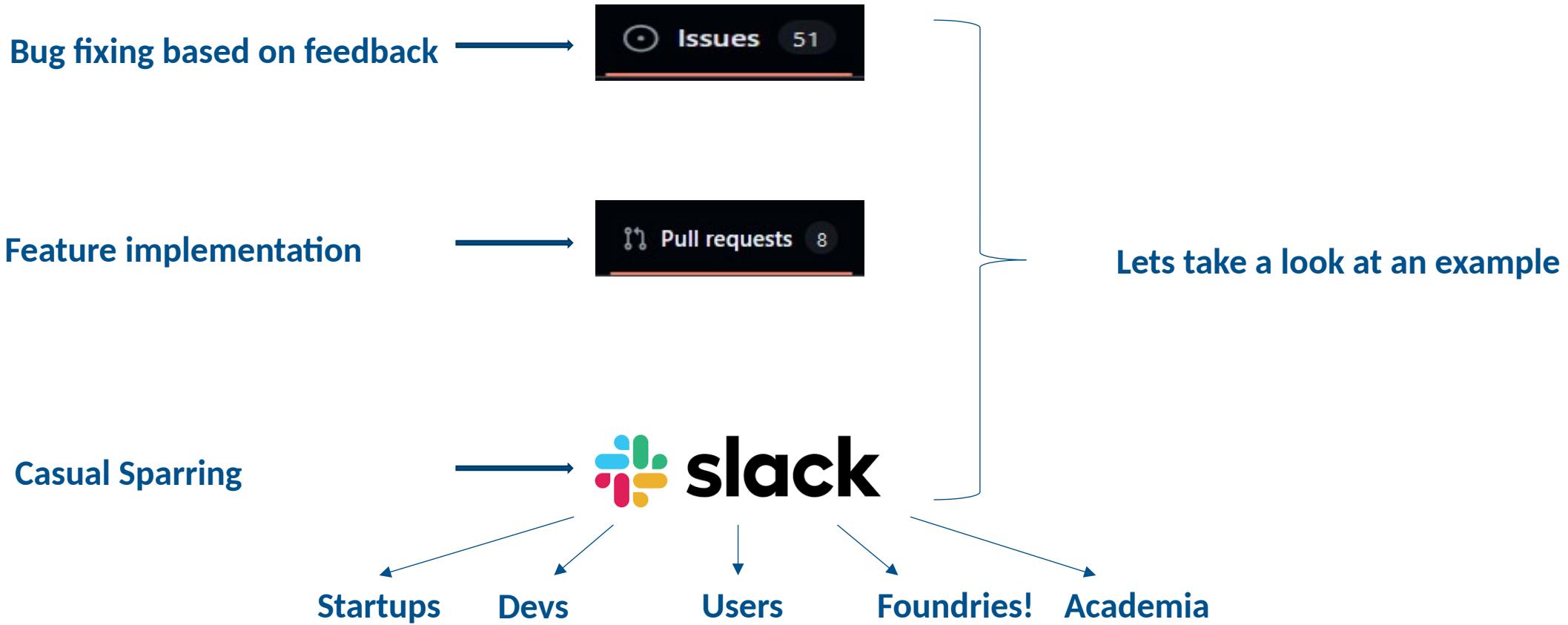
Understanding the power of the community

Planification of Open MPW runs

Leassons learned after three months and future activities



How Does the OS Community Aid the Development



Example on How to use the OS community

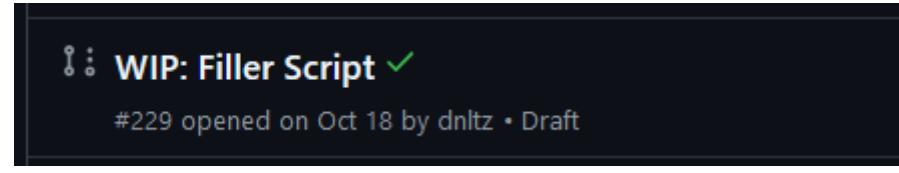


Opening an issue ticket on Github



Next steps?

Opening a pull request on Github



Next steps?

Doubts and Fears regarding the tools



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

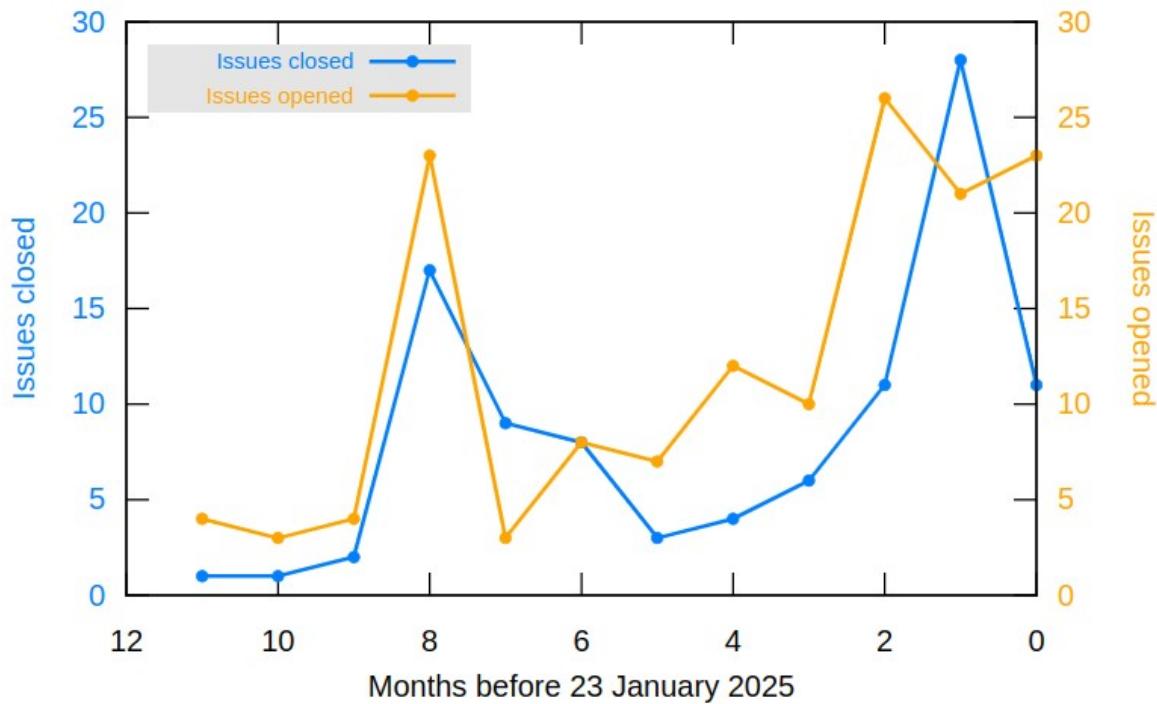


Enjoy!

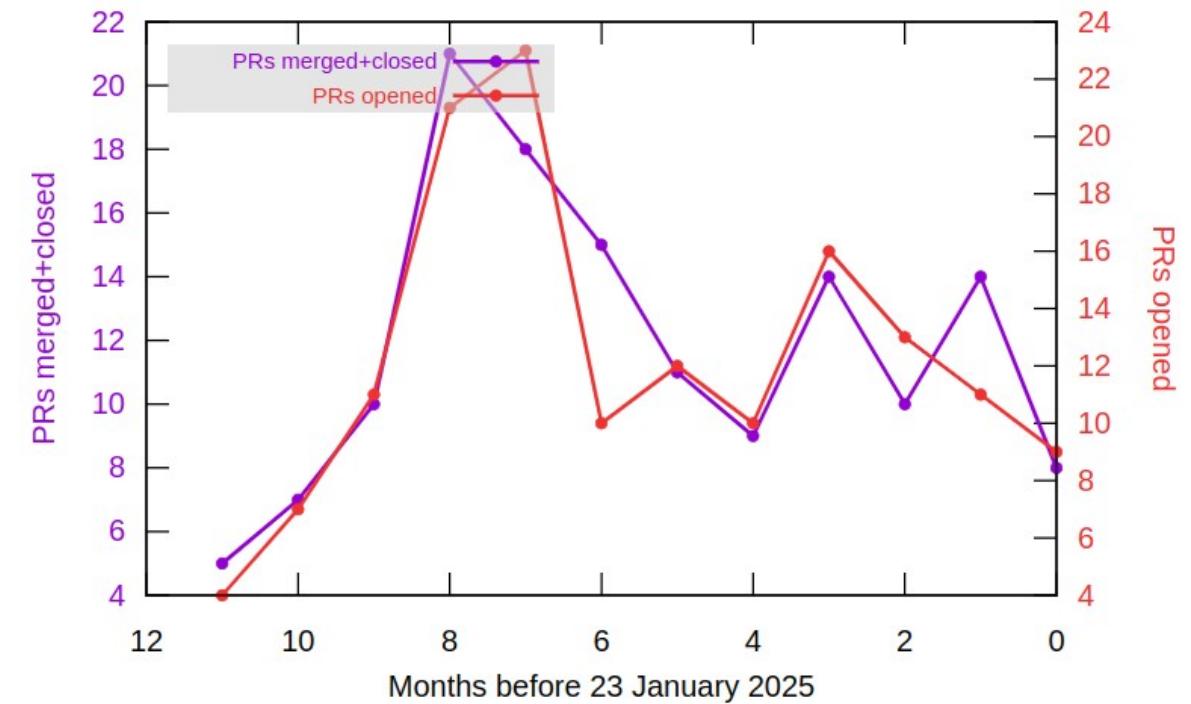
IHP-Open-PDK statistics



Issues closed and opened – IHP-GmbH/IHP-Open-PDK



PRs merged/closed and opened – IHP-GmbH/IHP-Open-PDK





Work in progress – analog/mixed/RF flow

- 0 Parasitics Extraction PEX – ongoing, <https://github.com/martinjankoheler/klayout-pex>
- 0 Noise modeling in ngspice (transient noise, low frequency noise) – ongoing
- 0 Qucs-S support – ongoing, agnostic PDK support, RF features
- 0 Device models – issues found and reported by community members, migration to Verilog-A behavioral models
- 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
- 0 Mixed mode testcases using xspice + verilator



Table of contents

Introduction to the IHP

Our motivation and goals in the open source silicon domain

Description of the open source soultions developed and supported by IHP

Understanding the power of the community

Planification of Open MPW runs

Leassons learned after three months and future activities

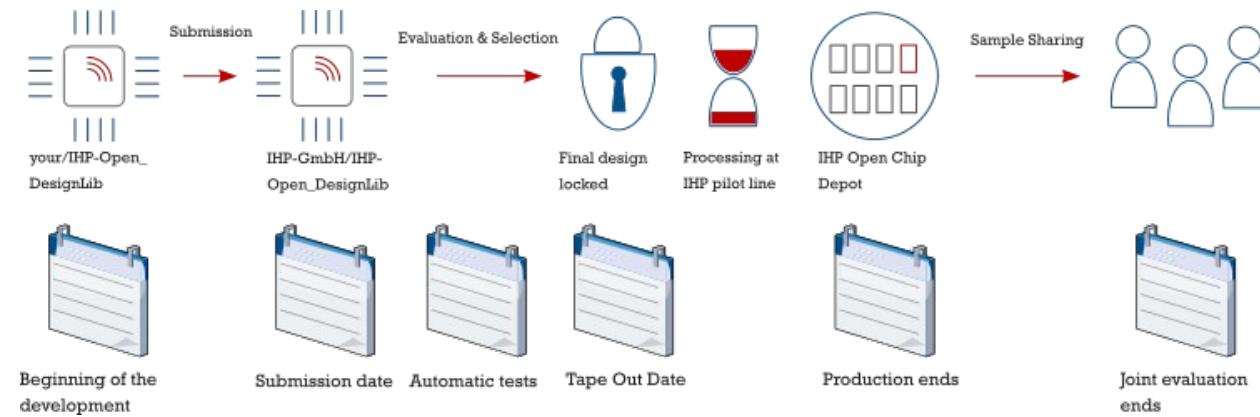


Free MPW Runs - support open source PDK & design

- 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



Table of contents

Introduction to the IHP

Our motivation and goals in the open source silicon domain

Description of the open source solutions developed and supported by IHP

Understanding the power of the community

Planification of Open MPW runs

Lessons learned after three months and future activities



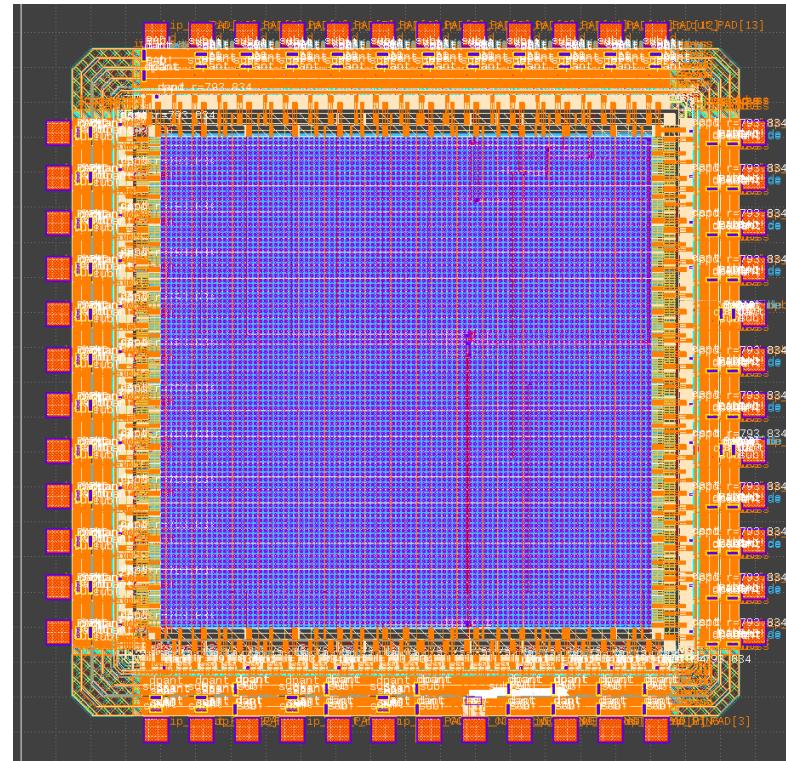
Key takeaways

- 0 Strong open-source community and interest (People are very nice!)
- 0 The open source enables and facilitates collaboration (consider an internship @IHP)
- 0 OS tools develops your understanding of semiconductor manufacturing etc.
- 0 Chip design demystification - hands on experiences
- 0 EDA tools well oriented to support open source PDK's, but still a lot to be implemented (get involved)
- 0 Open source Design library is still missing (get involved!)
- 0 Our team is growing, check out the open positions (ADK, PDK developers)

Full chip design example

- OpenROAD-flow-scripts does not provide full chip design example
- Padring design can be a tedious thing
- <https://github.com/KrzysztofHerman/orfs-design-template>

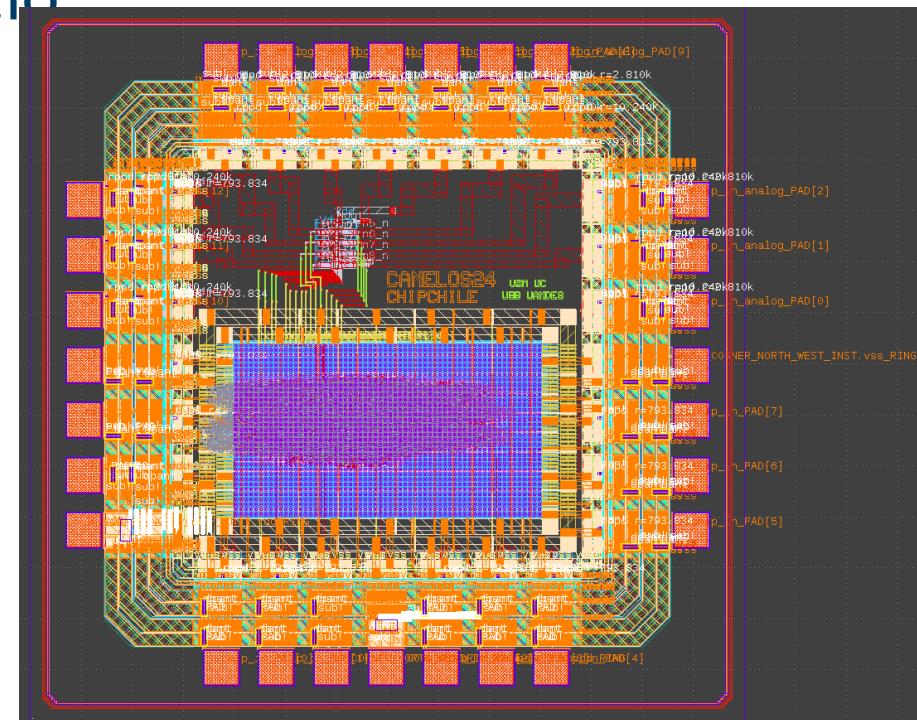
Cell / Category	Count (Not Visited) - Waived
By Cell	
[ihp_top]	
AFil.g	102 (101)
AFil.g2	102 (101)
M1.j	1 (1)
M2.j	11 (11)
M3.j	1 (1)
M4.j	1 (1)
M5.j	1 (1)
M1Fil.h	11 (11)
M2Fil.h	36 (36)
M3Fil.h	14 (14)
M4Fil.h	14 (14)
M5Fil.h	9 (9)
TM2.c	1
By Category	
All	102 (101)
	102 (101)



Mixed signal design using Ngspice



- NGSPICE is capable to handle a mixed mode simulation using shared object generated by Verilator from Verilog behavioral code.
 - <https://canelosworkshopihp2024-docs.readthedocs.io>
 - 8:1 analog mux
 - SPI interface
 - neural network





How to reach us

Email

-0 openpdk@ihp-microelectronics.com

GitHub

-0 issues – for reporting issues, questions

-0 discussions – for requesting features

Open Source Silicon Slack [channel](#): #ihp-sg13g2

-0 general discussions

-0 announcements

The image shows two screenshots. The top screenshot is of a GitHub repository page for 'IHP-GmbH / IHP-Open-PDK'. It displays the repository name, a code icon, an issues icon (labeled 'Issues 24'), a pull requests icon (labeled 'Pull requests 4'), and a discussions icon. The bottom screenshot is of an Open Source Silicon Slack channel named '#ihp-sg13g2'. It shows a list of channels including 'generative-ai', 'announcements', 'general', and 'ihp-sg13g2' (which is highlighted). A message from user 'Alex Sheldon' at 3:29 AM is shown, stating: 'joined #ihp-sg13g2.' Another message from the same user at 3:55 AM says: 'I am just getting set up to simulate with this process. However it seems like no matter what I change on with W=1um and L=0.45um. Any ideas here? its my first time using the tools so sure I am doing something wrong'. Below the messages is a screenshot of a CAD tool interface showing a circuit diagram and some command-line text related to GDS3D viewer configuration.

Acknowledgment



- o Thanks to my colleagues at IHP and all the people involved in the development of the PDK
- o And special thanks to the following public founded German projects:
 - o VE-HEP (16KIS1339K) <https://elektronikforschung.de/projekte/ve-hep-1>
 - o IHP Open130-G2 (16ME0852)
<https://www.elektronikforschung.de/projekte/ihp-open130-g2>
 - o FMD-QNC (16ME0831) <https://www.elektronikforschung.de/projekte/fmd-qnc>

SPONSORED BY THE



Federal Ministry
of Education
and Research





Thank you for your attention!

IHP GmbH - Innovations for High Performance Microelectronics

Im Technologiepark 25

15236 Frankfurt (Oder)

Tel.: +49 (0) 335 5625 380

E-Mail: herman@ihp-microelectronics.com

