

Open-source process design kit based on IHP-SG13G2 technology

Krzysztof Herman



HeiChips, Heidelberg 2025

Projects: BMFTR -> FMD-QNC (16ME0831)

About me



- BSc in electronics
- MSc in Acoustics
- PhD in Telecommunications
- 15 years of experience in academia Poland/Chile
- bi-polar experience
- IHP-Open-PDK developer since 2023

...one of the first „victims” of the open-source silicon movement

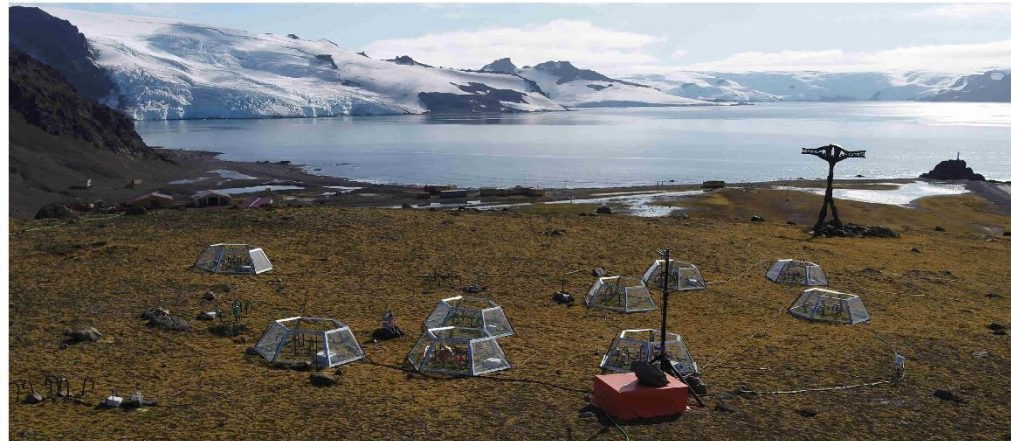


Table of contents



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

How do we make the PDK ? ?

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		
HeiChipsors	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

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

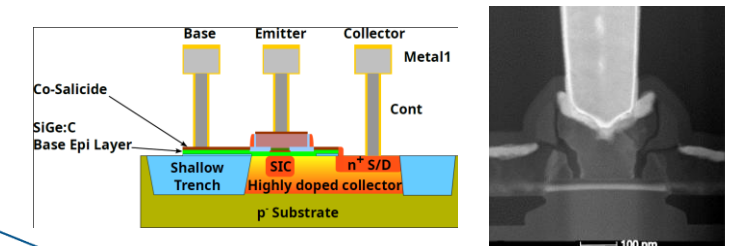


Table of contents



Introduction to the IHP

Our motivation and goals in the open source silicon domain

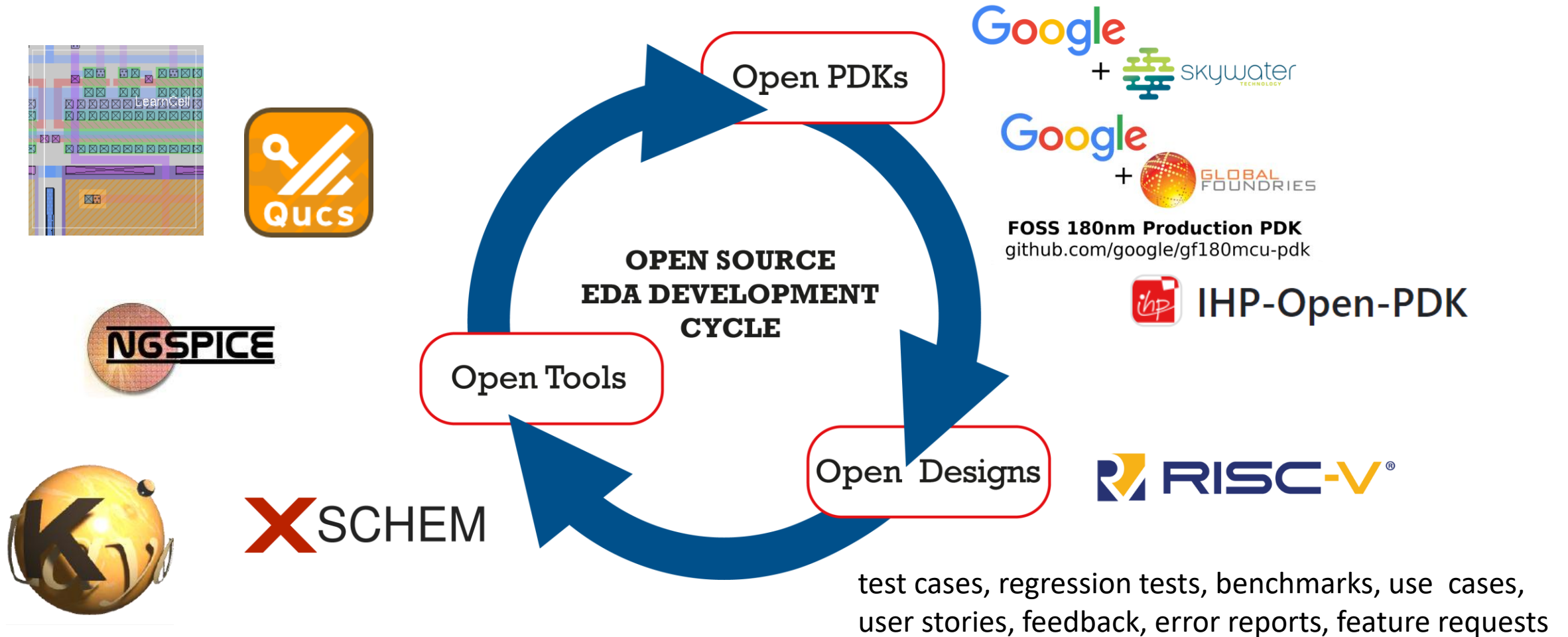
Description of the open source solutions developed and supported by IHP

Understanding the power of the community

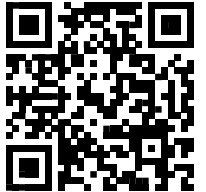
Planification of Open MPW runs

How do we make the PDK ? ?

Open source ASIC development cycle



IHP Open PDK on github



IHP-GmbH / IHP-Open-PDK

Search: Type / to search

Issues 135 Pull requests 19 Discussions Actions Projects 1 Wiki Security Insights Settings

IHP-Open-PDK Public

Edit Pins Unwatch 30 Fork 96 Star 560

dev 2 Branches 2 Tags

Go to file Add file Code

This branch is 162 commits ahead of main

KrzysztofHerman Merge pull request #559 from FlinkbaumFAU/remove_verilogp... 1 hour ago 961 Commits

.github	Updating layout installation for actions	5 months ago
ihp-sg13g2	Merge pull request #559 from FlinkbaumFAU/remove_verilo...	1 hour ago

About

130nm BiCMOS Open Source PDK, dedicated for Analog, Mixed Signal and RF Design

ihp-open-pdk-docs.readthedocs.io

open-source pdk ihp

Readme

Minimal installation – plug and play & KISS approach

Working with 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 source code usually available on platforms like github, gitlab, sourceforge 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 necessary 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
```

Ubuntu 22.04+ compatible
Configuration: \$PDK_ROOT, \$PDK
Dependencies installation: mainly apt-get
Tools have to be installed (binaries, sources)



Getting started really isn't that hard!



```
pedersen@IHP-OPDK: ~  
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

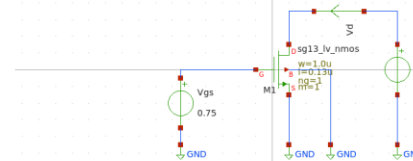
- dc_lv_nmos x5
- dc_hv_nmos x6
- dc_lv_pmos x7
- dc_hv_pmos x8
- dc_mos_temp x11
- dc_mos_cs_temp x12
- dc_res_temp x13
- dc_ntap1 x25
- dc_ptap1 x26
- dc_diode_op x14
- dc_diode_temp x15
- dc_hbt_13g2 x17
- dc_pnpMPA x30
- dc_logic_not x28

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
```



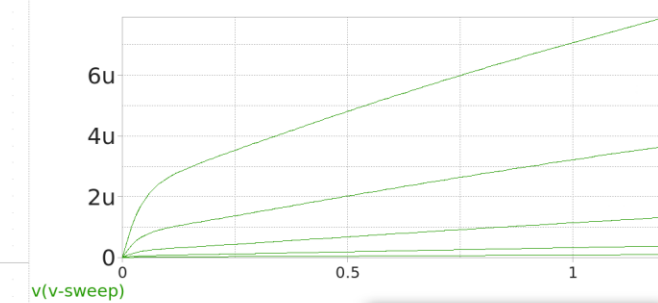
MODEL

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



Copyright 2023 IHP PDK Authors
x5. sg13g2_tests/dc_lv_nmos.sch

i(vd)



load waves Ctrl + left click

```
dc_lv_nmos.spice" -a || sh  
*** Copyright 2003-2004. The ngspice team.  
*** Please get your ngspice manual from https://ngspice.sourceforge.io/docs.html  
*** Please file your bugreports at https://ngspice.sourceforge.net/bugrep.html  
*** Creation Date: Tue Jul 16 09:23:55 UTC 2024  
*****  
Notes: No compatibility mode selected!  
Circuit1 == sg13g2_tests/ihp-open-pdk/ihp-sg13g2/libs.tech/xschem/sg13g2_tests/dc_lv_nmos.sch  
Doing analysis at TEMP = 27.00000 and TNOM = 27.00000  
Using SPARSE 1.3 as Direct Linear Solver  
No. of Data Rows : 1  
Doing analysis at TEMP = 27.00000 and TNOM = 27.00000  
Using SPARSE 1.3 as Direct Linear Solver  
No. of Data Rows : 605  
Writing raw file "dc_lv_nmos.raw"  
ngspice GO ->
```

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



Design examples and courses

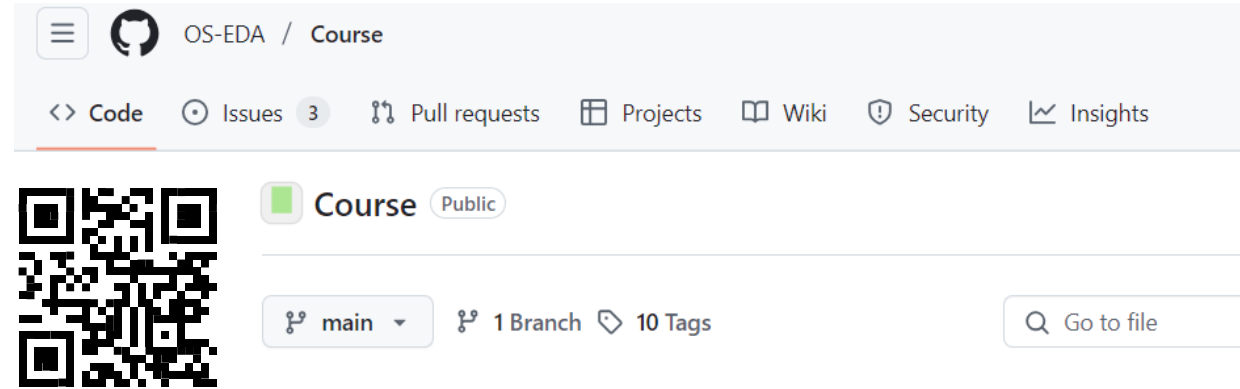
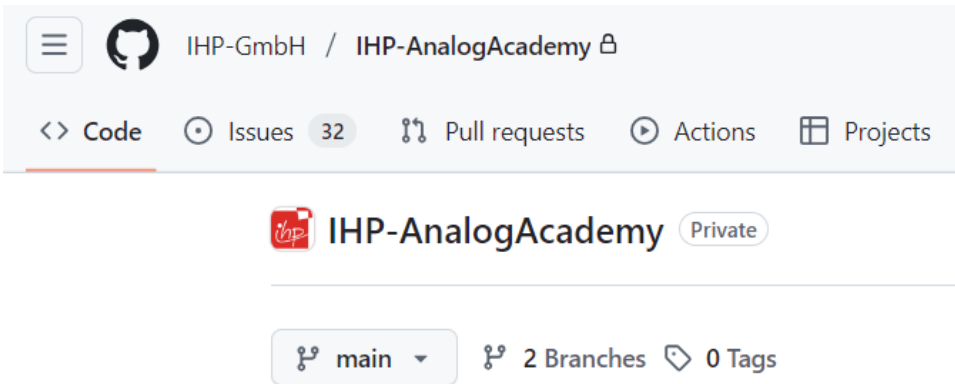


Table of contents

- [Introduction to IHP Open PDK and SG13G2 Technology](#)
- [Foundations](#)
- [Bandgap Reference](#)
- [50GHz Medium Power Amplifier](#)
- [8-bit SAR ADC](#)
- [Final Thoughts](#)

Course instance: 1 Week, hands on

Mon	Tue	Wed	Thu	Fri
L1: Introduction	Q1, Q2: Recap Feedback	Q3, Q4: Recap Feedback	Q5, Q6: Recap Feedback	Q7: Recap
T1: Training	L3: Verilog T3: Training	L5: PDK T5: Training	L7: OpenROAD Flow scripts T7: Training	L8: Tapeout Feedback
L2: OpenROAD tools	L4: OpenROAD first run	L6: OpenROAD GUI	L7: OpenROAD Flow scripts 2	Spare time and Wrap-Up
T2: Training	T4: Training	T6: Training	T7: Training	

L : Lectures
T : Training and
Hands-On
Q : Questions

Table of contents



Introduction to the IHP

Our motivation and goals in the open source silicon domain

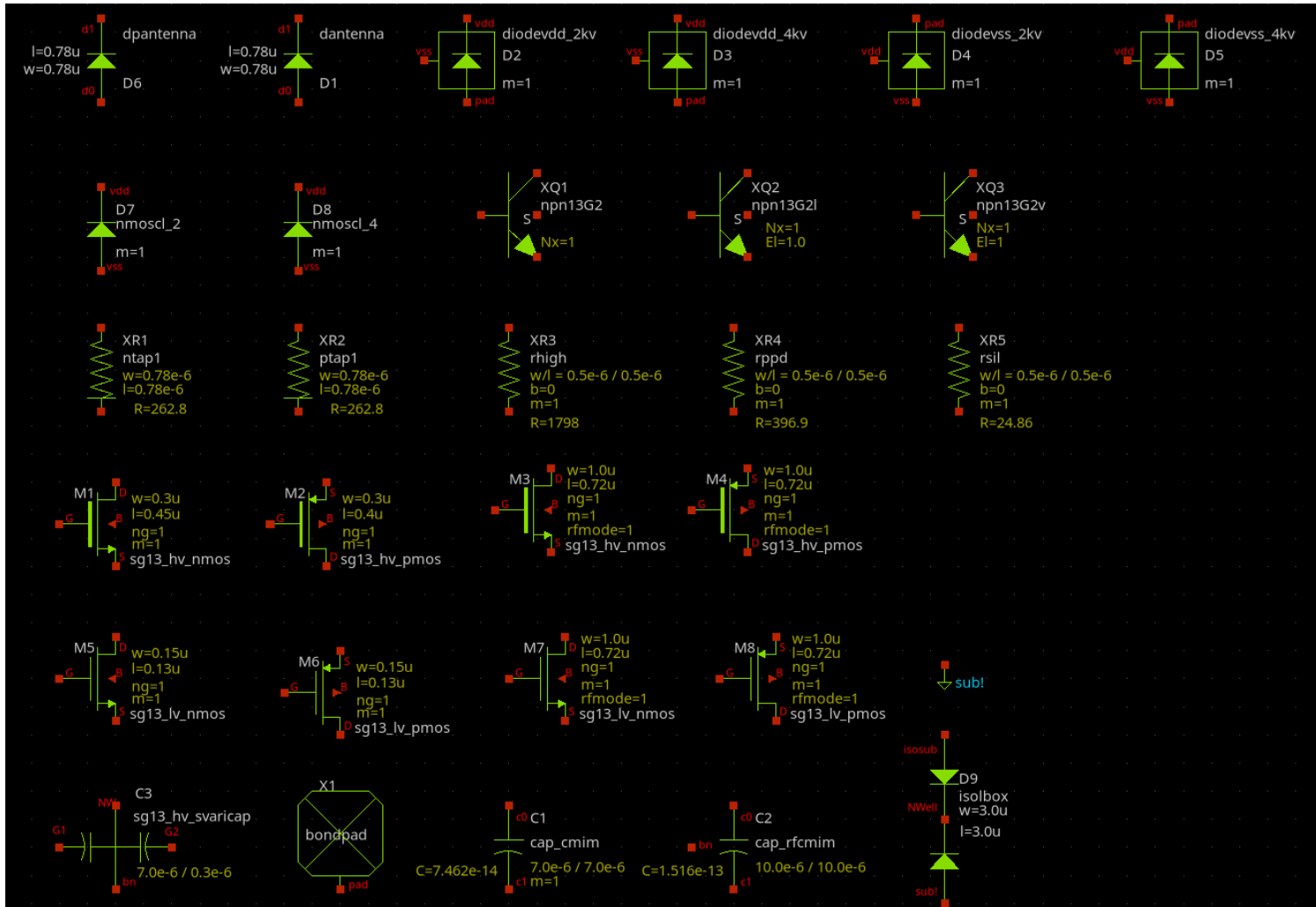
Description of the open source solutions developed and supported by IHP

Understanding the power of the community

Planification of Open MPW runs

How do we make the PDK ? ?

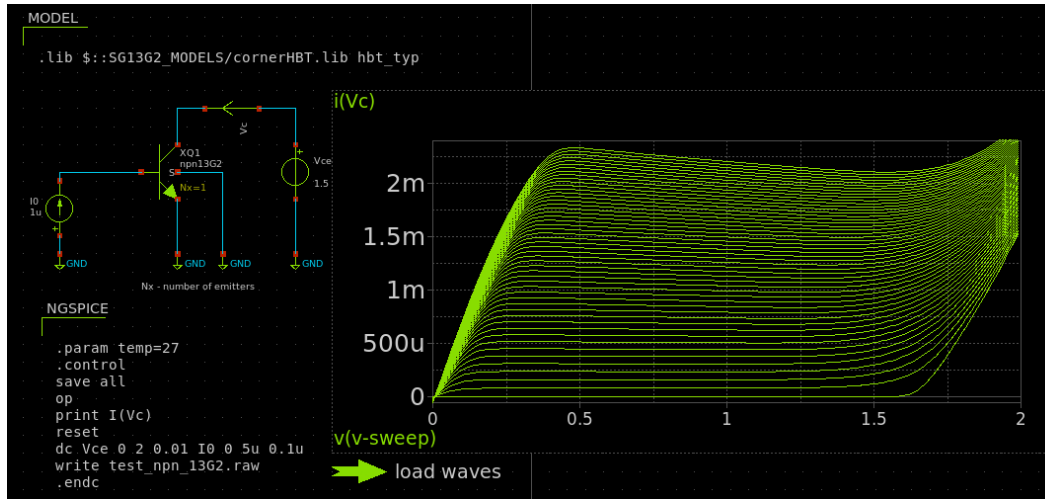
OpenPDK support for schematic capture



The current version of the IHP OpenPDK supports:

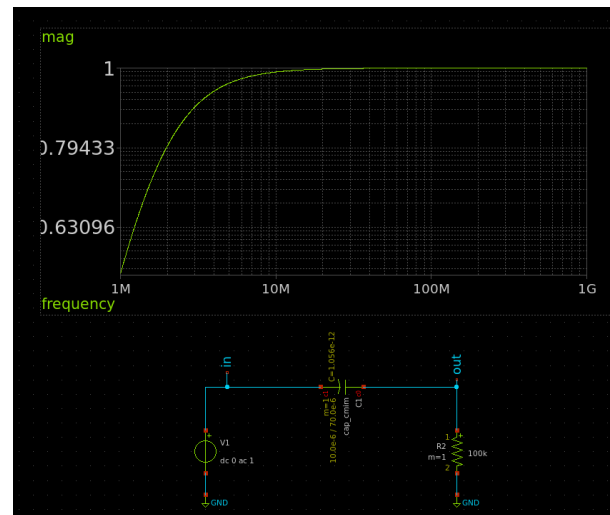
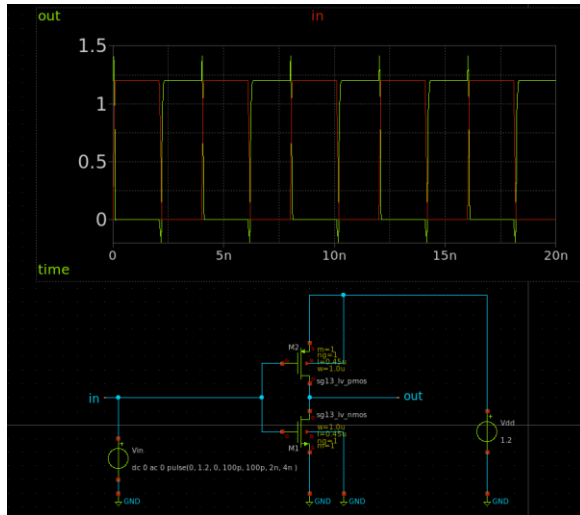
- xschem primitives for schematic capture
- automatic ngspice/Xyce compatible netlist generation
- example use cases to show the basic functionalities and parameters of the primitives

OpenPDK support for simulations

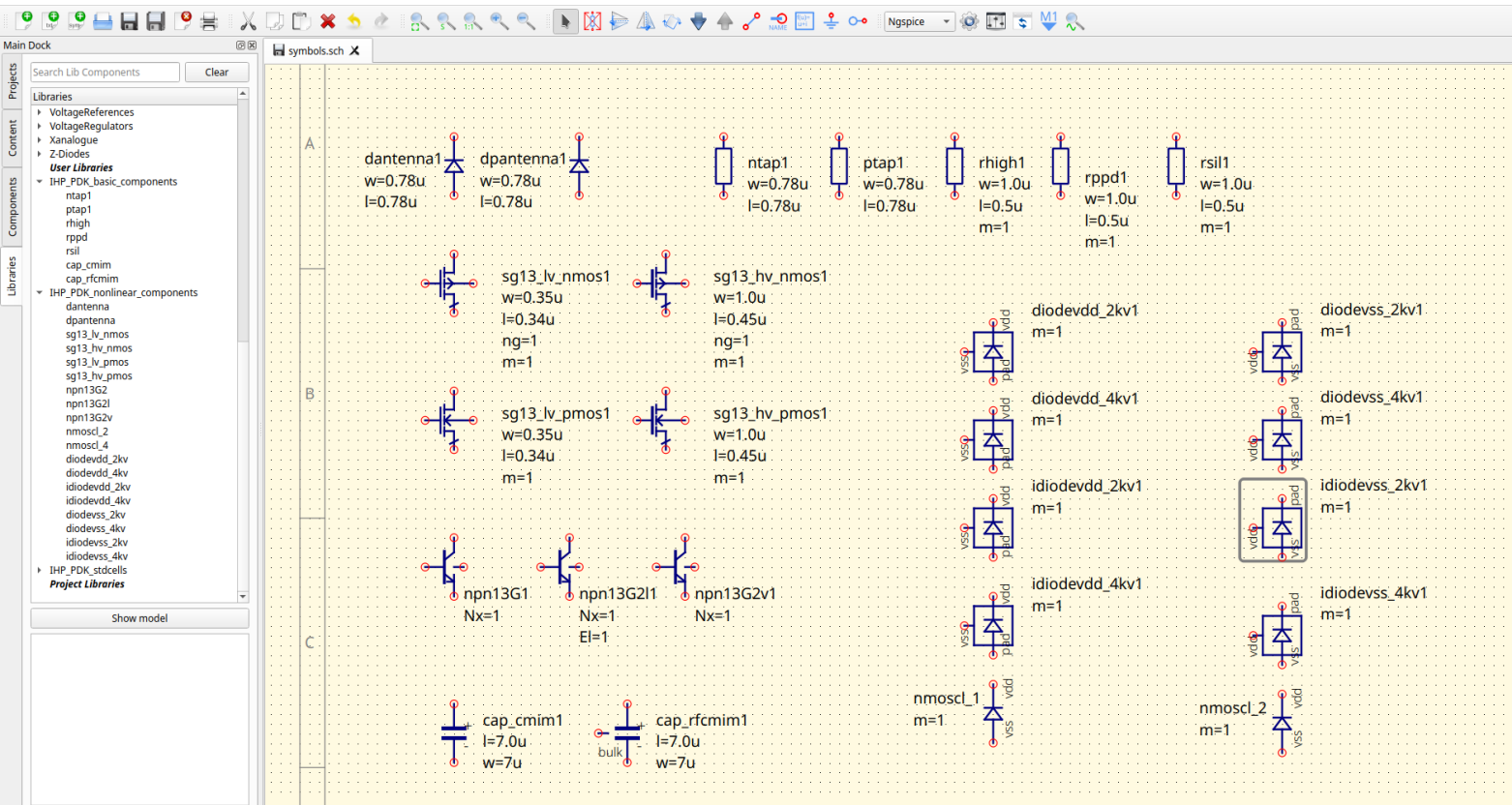


Analog simulation facilities:

- 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



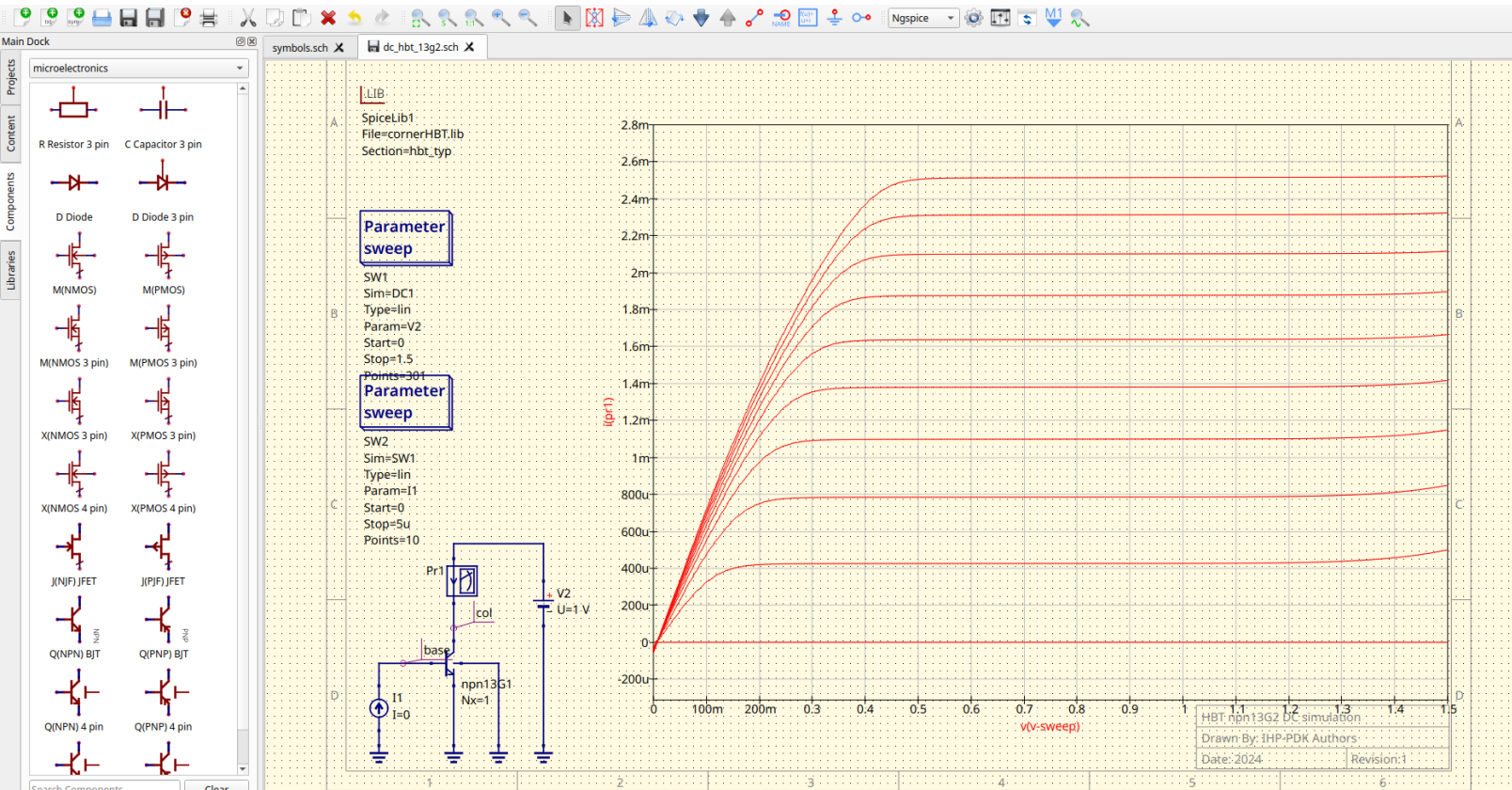
OpenPDK support for Qucs-S schematic capture



The current version of the IHP OpenPDK supports:

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

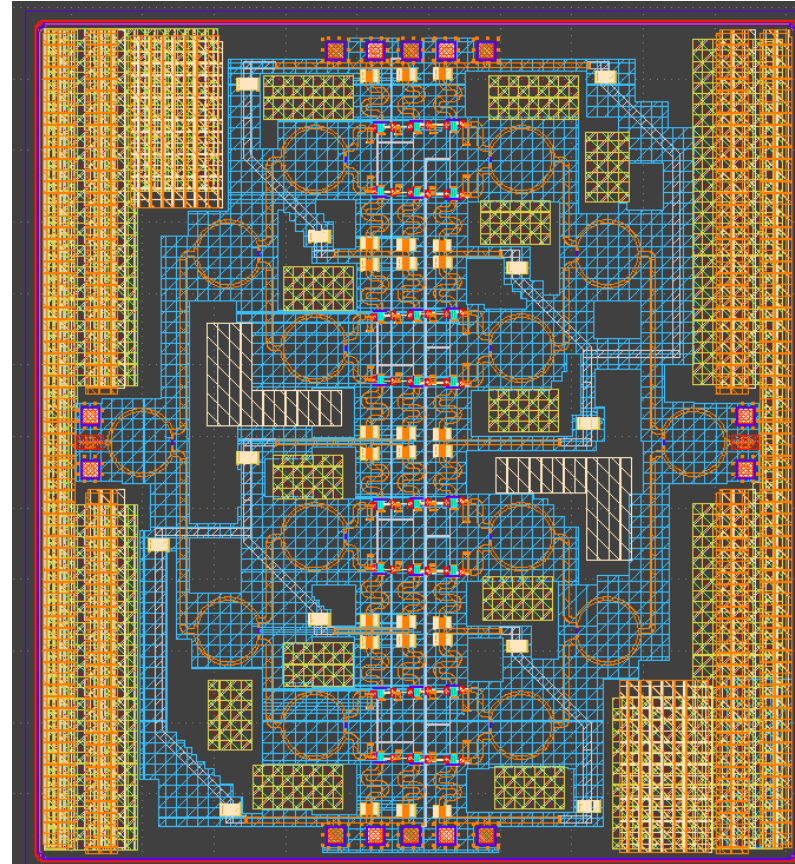
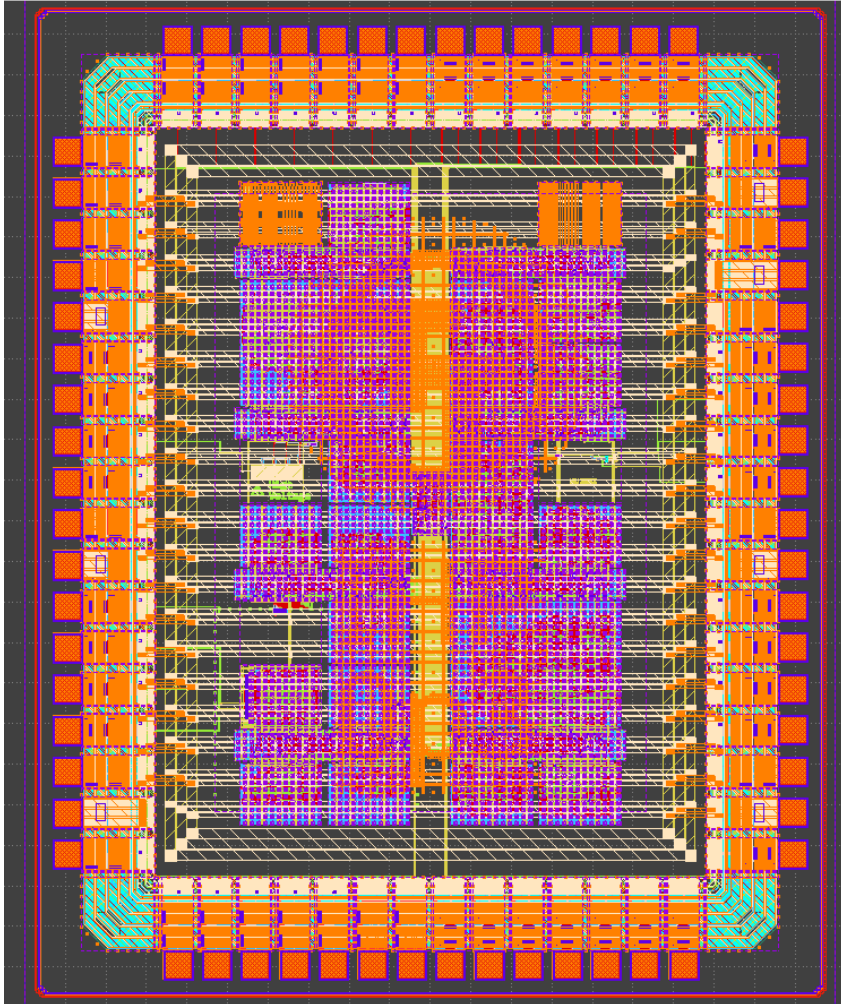
OpenPDK support for Qucs-S schematic capture



Qucs-S development:

- Agnostic support for PDK
- CDL netlisting
- Interoperability with OpenEMS and Klayout
- XML symbol library import
- RF related features

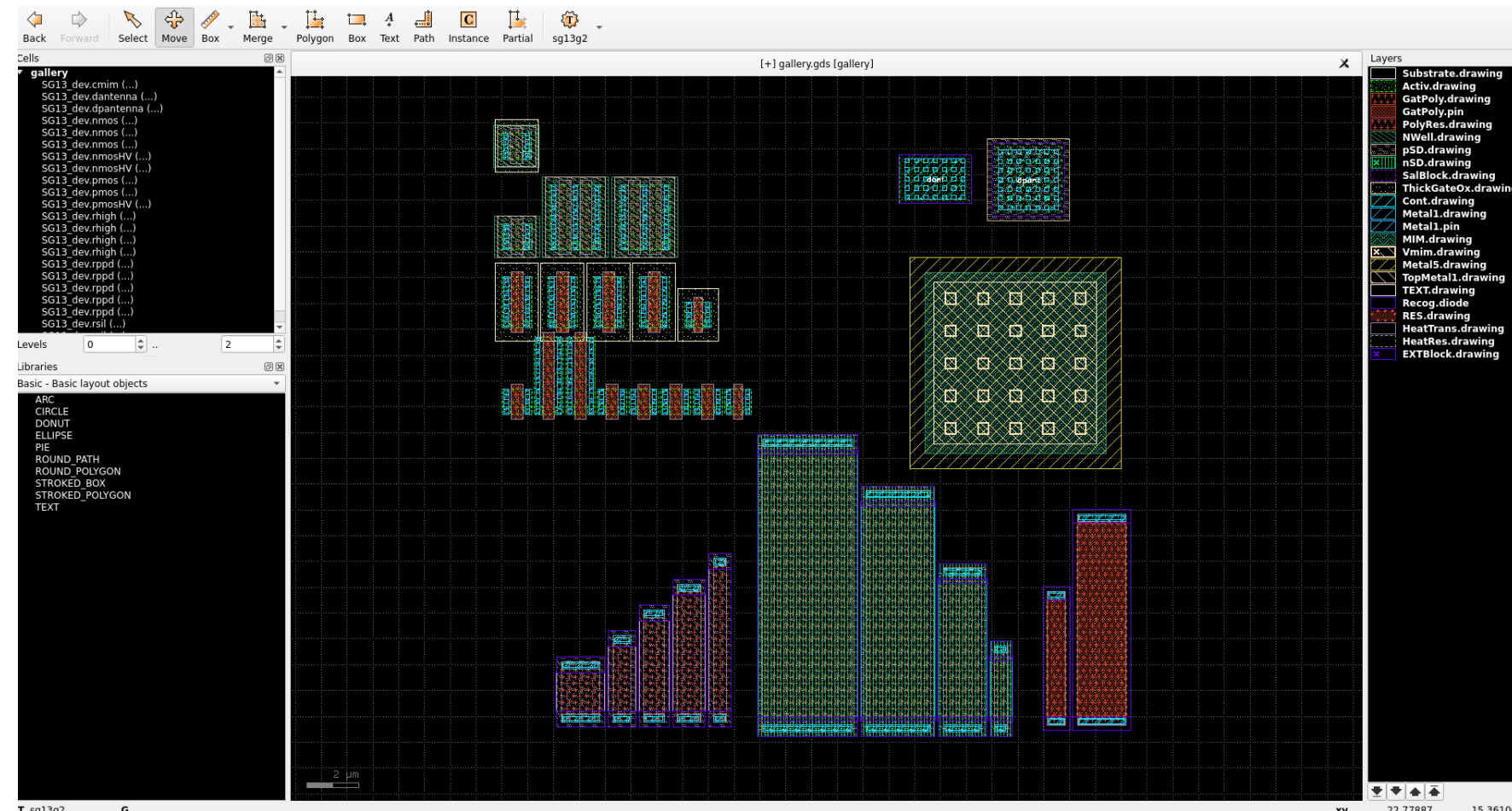
KLayout – primary tool for open source layout design



KLayout key features:

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

KLayout – Pycell support



- Pycells compatible with Synopsys PyCell Studio and Keysight ADS
- KLayout wrapper API development
- Klayout Device generator

```
klayout -n sg13g2 -zz -r generator.py -rd netlist=netlist.spice -rd output=macros/gallery.gds
```

KLayout example DRC run on `gatpoly` QA cell



KLayout 0.28.12 - sg13g2_qacells.gds [gatpoly]

File Edit View Bookmarks Display Tools Macros Help

Back Forward Select Move Ruler sg13g2

Cells

- gatpoly
- metal1
- nwell

Layers

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

Marker Database Browser

Database: sg13g2.lyrdb
... on layout: sg13g2_qacells.gds

Directory

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)

Markers

F	I	W

Info ☒ list shapes

Gat.d [gatpoly]

Min. GatPoly to Activ space = 0.07

Configure Close

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]	
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:

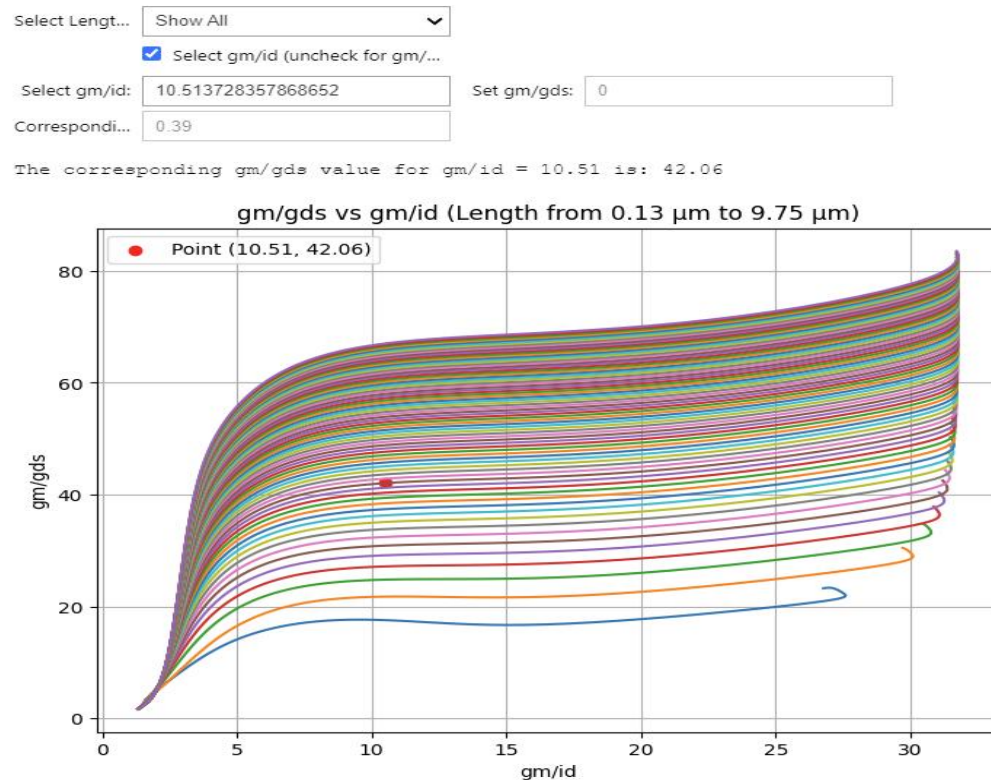
- accepts CDL netlist,
- extracts devices from layout,
- creates extracted netlist,
- performs checks and compares:
 - Pins
 - Nets
 - Devices
- reports inconsistencies
- can run in batch mode

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

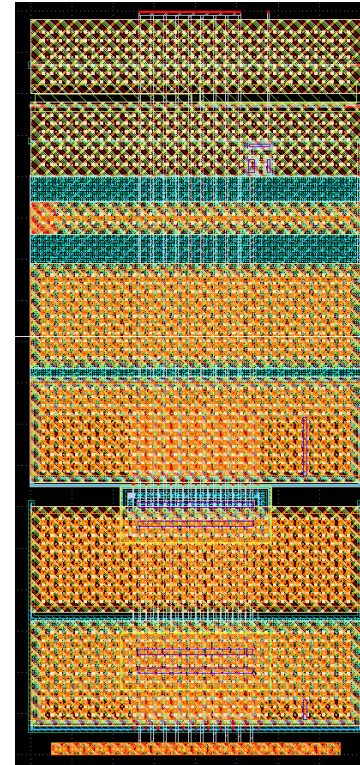
Layout Automation

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

Digital primitives

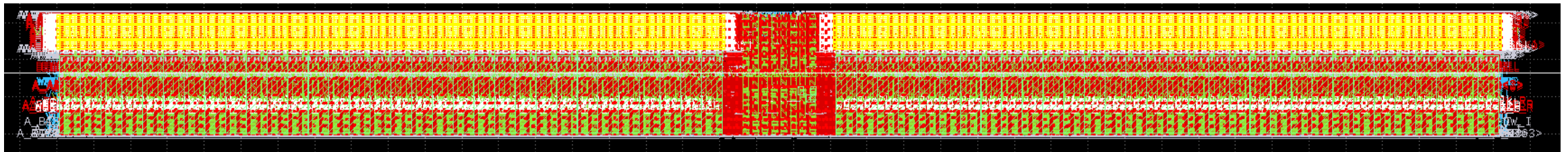


	StdCells	IOCells	SRAM
Verilog	x	x	x
LIB	x	x	x
LEF	x	x	x
CDL	x	x	x
SPICE	x	x	x
GDS	x	x	x

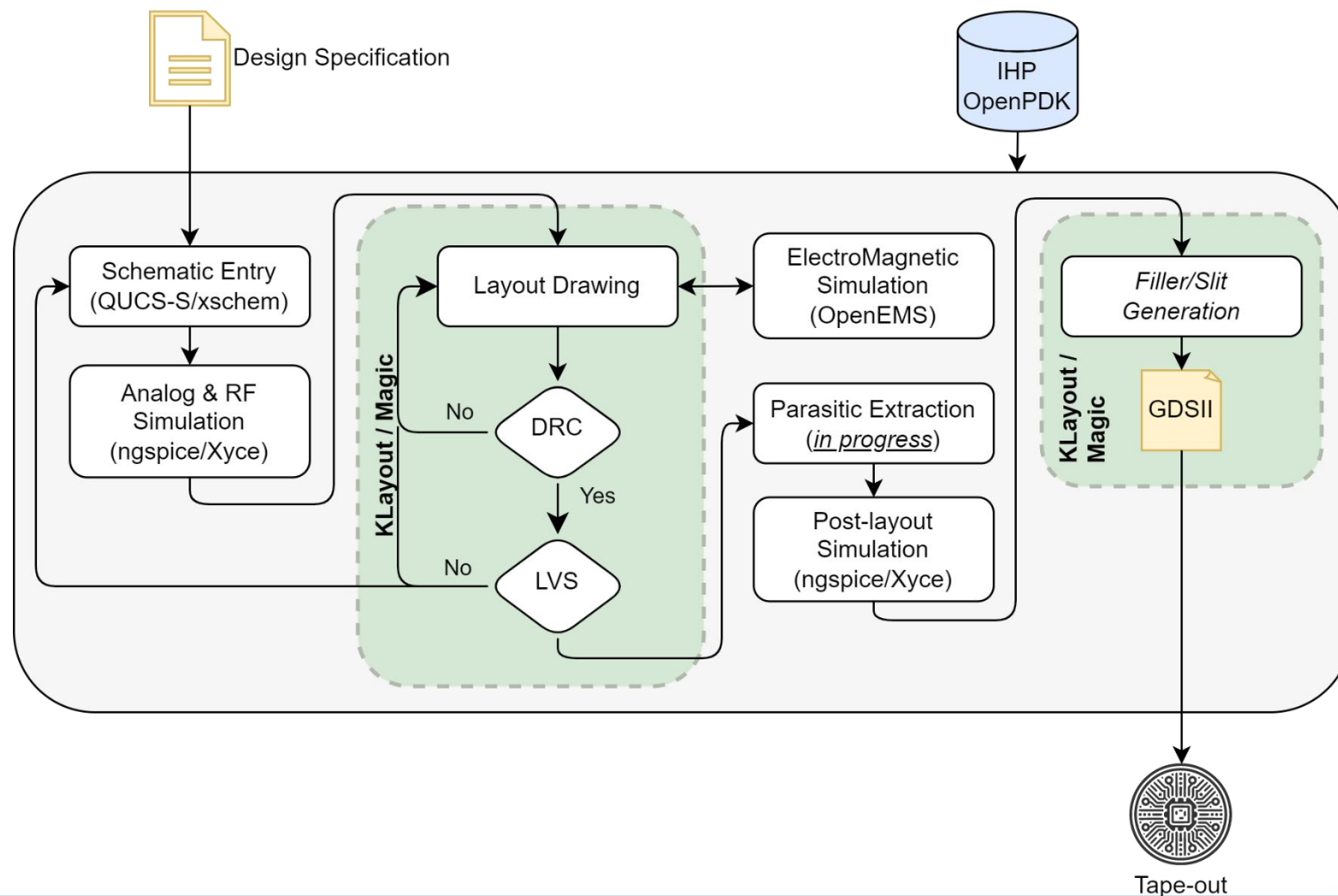


Digital components:

- 0 stdcells
 - 0 84 cells,
 - 0 combinatorial logic,
 - 0 sequential elements,
 - 0 scan flops,
 - 0 gated cells.
- 0 IOCells,
 - 0 In, Out, InOut, Analog
 - 0 different drive strengths
- 0 SRAM
 - 0 hard macros of a single port SRAM
 - 0 different sizes

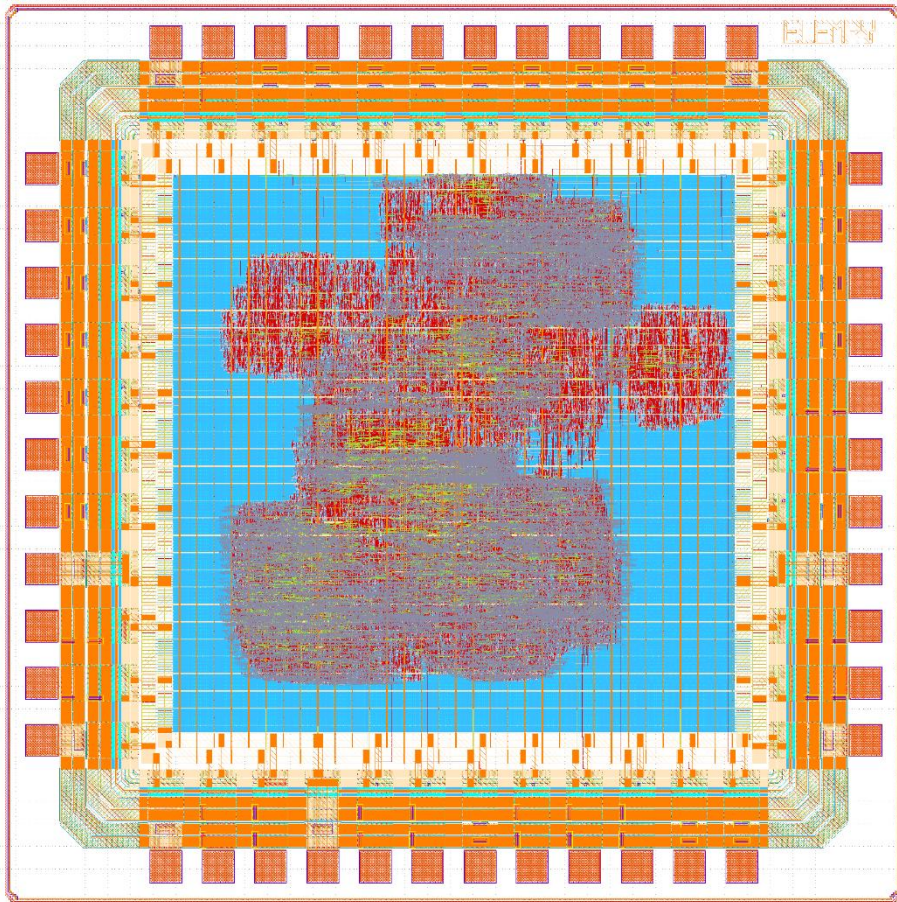


Analog/RF Open Source Design Flow: status at a glance

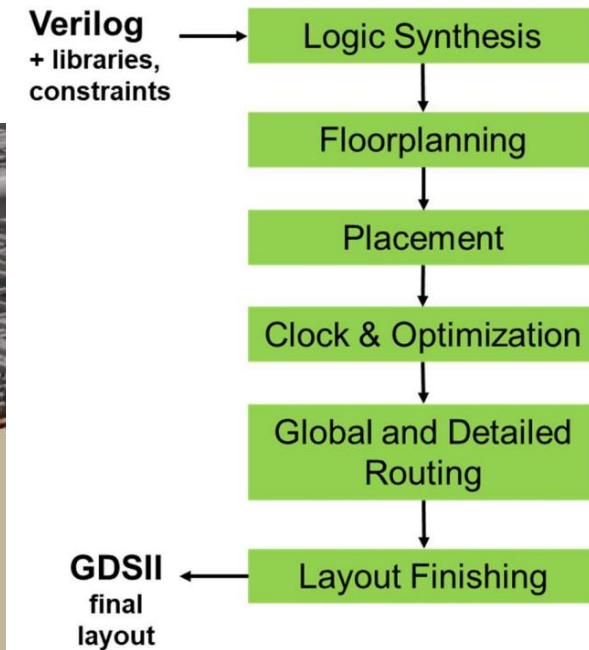
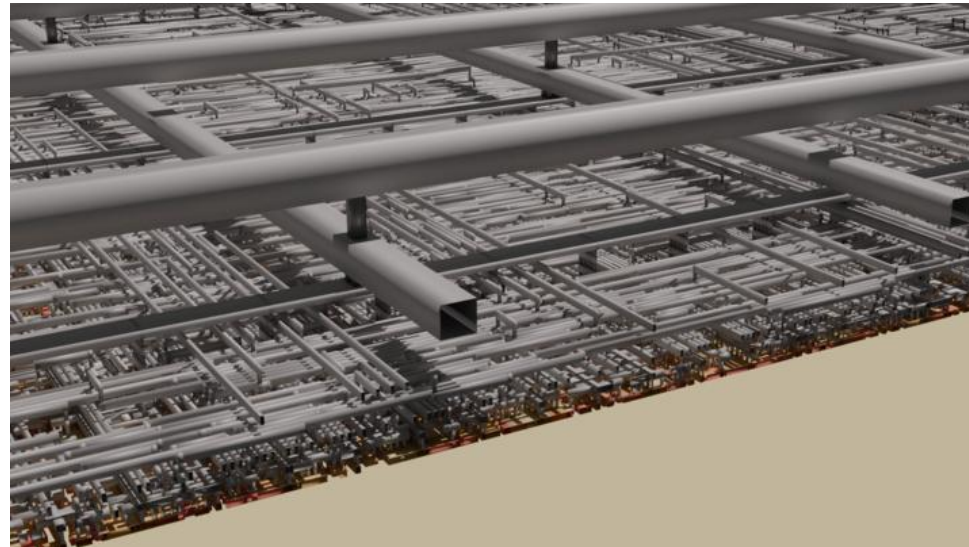


- KLayout-oriented flow
 - Layout design
 - Parameterizable cells
 - Physical Verification
- QUCS-S, xschem
- ngspice, Xyce
- OpenEMS, ElmerFEM, AWS Palace
- Working on a faster solver for EMS (ELMER)

Digital OpenROAD-flow-scripts Flow

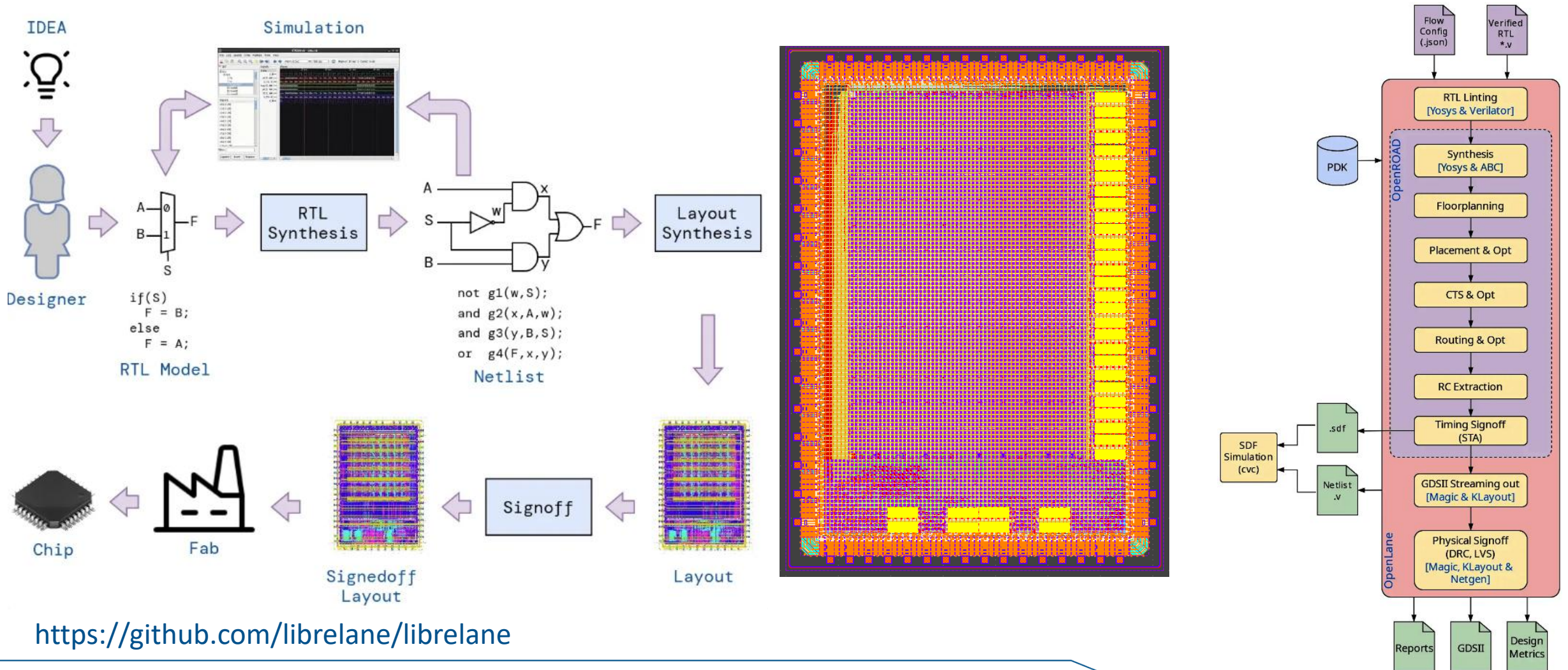
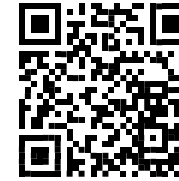


- Yosys + ABC -> synthesis
- OpenROAD -> PnR
- Klayout -> GDS streaming+checks



<https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts>

Digital LibreLane flow



<https://github.com/librelane/librelane>

Table of contents



Introduction to the IHP

Our motivation and goals in the open source silicon domain

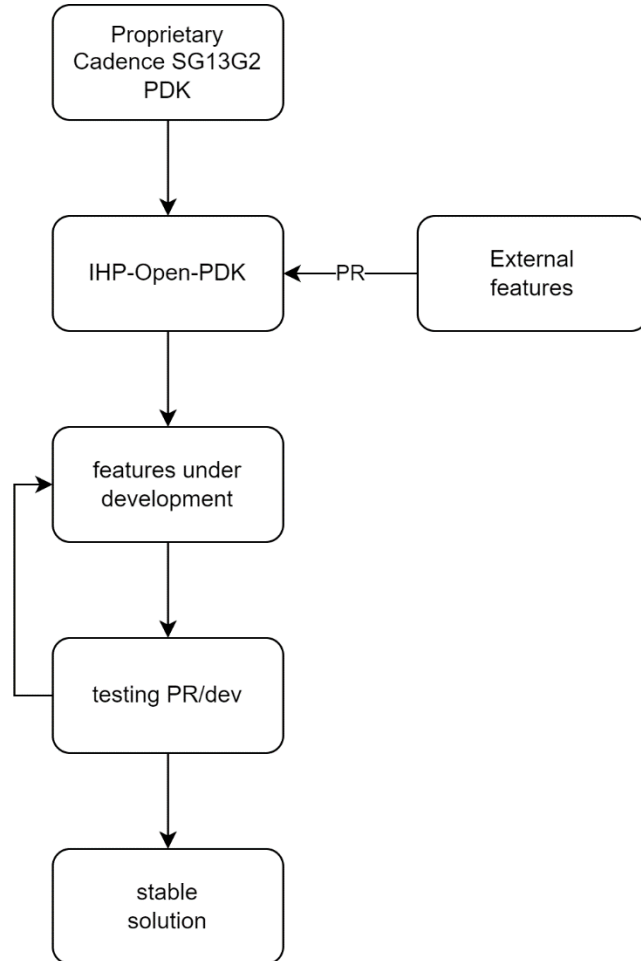
Description of the open source solutions developed and supported by IHP

Understanding the power of the community

Planification of Open MPW runs

How do we make the PDK ? ?

PDK development cycle

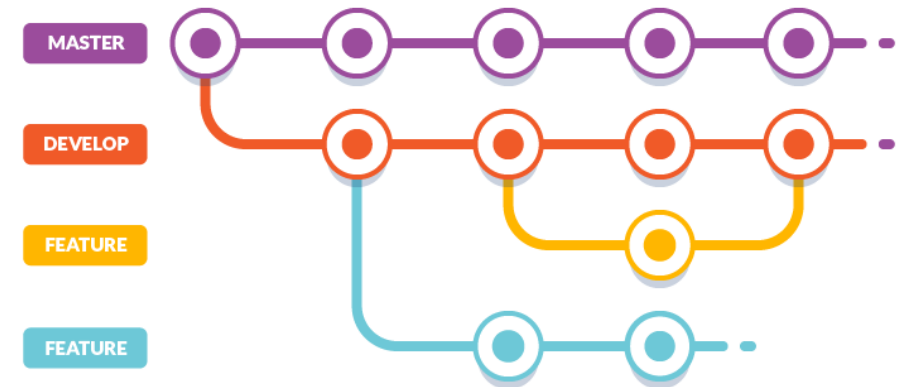


Principal rules

- Tight link between SG13G2 PDK and its open-source equivalent
- Compatibility with the open-source EDA tools
- Preserving the structure proposed in SKY130
- Read-only
- Based on git and following git flow

Key actors

- PDK core team
- Tools and flow developers
- Designers
- OS Community



PDK development cycle – community support



Issues 135 Pull requests 24 Discussions Actions Projects 1 Wiki

is:issue

☐ Open 135 ☐ Closed 165 Author ▾ Labels

- ☐ ☒ **Cannot run simulation on test scheme**
#566 · by thonglinh90 was closed yesterday
- ☐ ☒ **Questions about sealing For MPW shuttle runs** question
#562 · EngGhaith opened 4 days ago
- ☐ ☒ **Xyce plugins support limited to only 2 plugins** bug
#560 · KrzysztofHerman opened 4 days ago
- ☐ ☒ **Double prefix in stdcell names in verilog netlist created by xschem** bug
#557 · by FlinkbaumFAU was closed 4 days ago
- ☐ ☒ **Discrepancy in LVS Netlist Extraction for Parallel Capacitors**
#555 · PhillipRambo opened last week

Filters ▾

☐ ☒ 24 Open ☒ 231 Closed Author ▾

- ☐ ☒ **Code changes fixes the problem that Xyce/ADMS can't handle switch...** ✗
#568 opened 10 hours ago by dwarning
- ☐ ☒ **Adding some updates for LVS run setup and actions** ✓
#567 opened yesterday by FaragElsayed2
- ☐ ☒ **Stdcell update 2. Hot Fix 1.** ✓
#565 opened 3 days ago by b10346
- ☐ ☒ **Changes in Act and GatPoly filler macro.** ✓
#564 opened 3 days ago by KrzysztofHerman
- ☐ ☒ **Added bulk node to resistor models rhigh + rppd + rsil cont.** ✓
#563 opened 3 days ago by KrzysztofHerman • Draft
- ☐ ☒ **Xyce plugins sourced from a plugin list** ✓
#561 opened 4 days ago by KrzysztofHerman • Draft

Work in progress – analog/mixed/RF flow



- 0 Parasitics Extraction PEX – ongoing, <https://github.com/martinjankoehler/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 New and fast DRC ruledeck for klayout
- 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 solutions developed and supported by IHP

Understanding the power of the community

Planification of Open MPW runs

How do we make the PDK ? ?

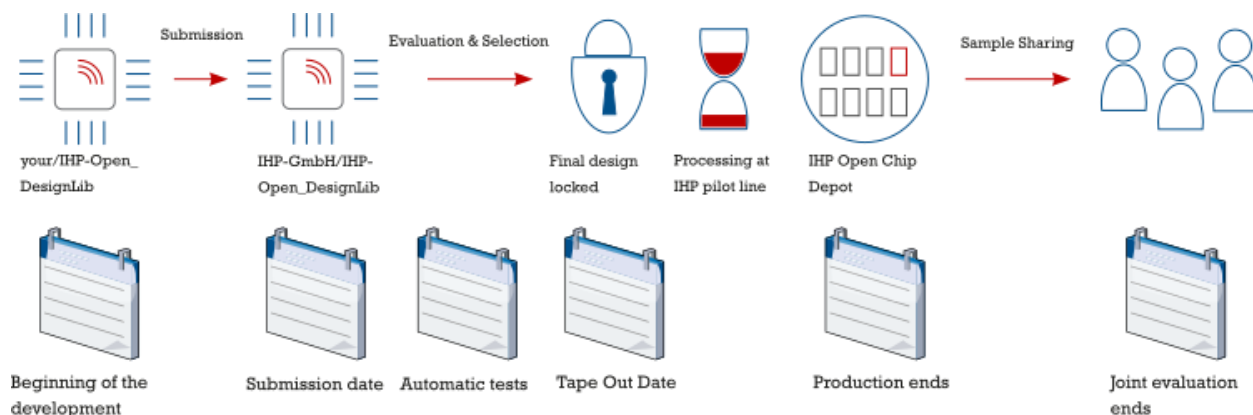
Free MPW Runs - support open source PDK & design



0 The table provides schedule of MPW Runs for FMD-QNC project in 2025 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/>

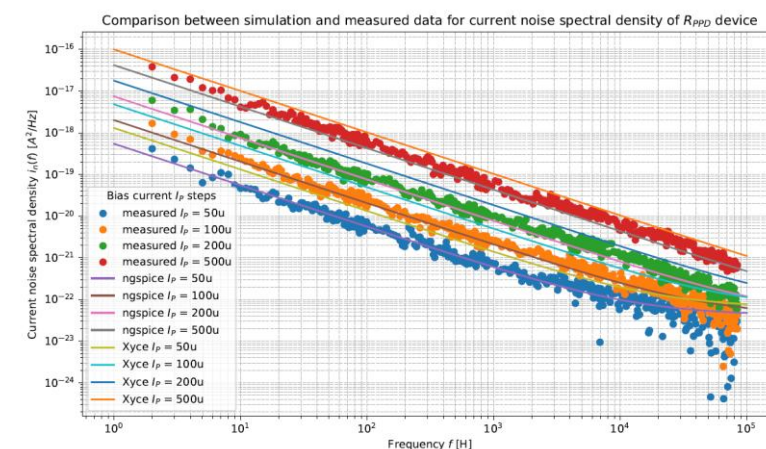
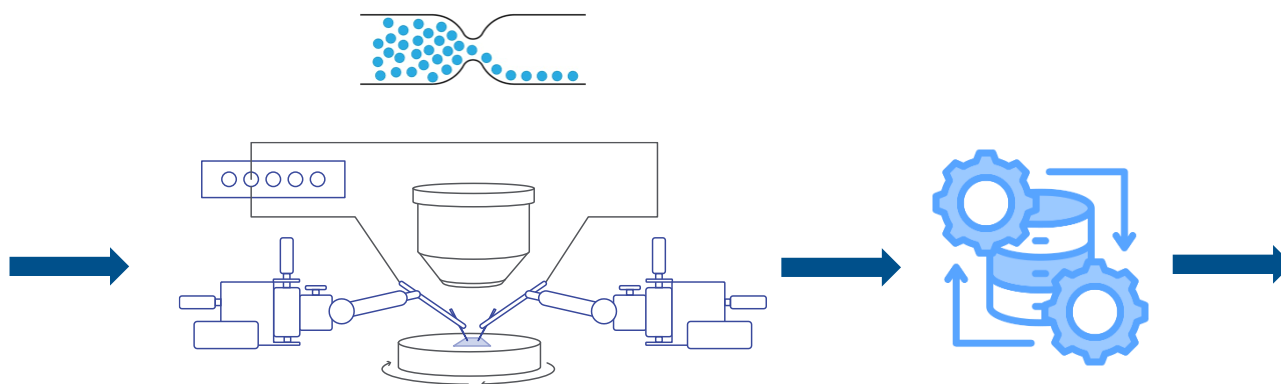
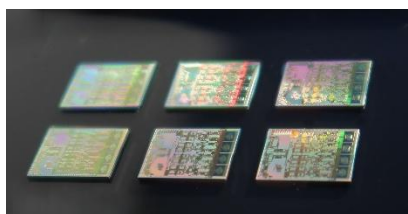
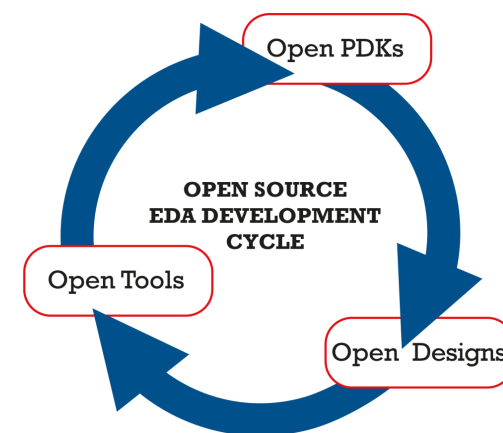


- 0 Project funds can be used exclusively to produce chip designs for non-economic activities, such as university education, research projects, and others
- 0 A concept for sustainable provision of free or low-cost MPW area for the open source community is to be developed.

The future – Joint Evaluation



- Receiving your own silicon is just one milestone—it doesn't mean the development cycle is complete.
- The samples/wafers from free OpenMPW will be stored at IHP
- Anybody can rent chips for measurements
- Silicon cross validation as research good practices
- Verified IP can become part of an open-source design library
- **How to organize the measurement to balance the load of the measurements lab ?**
- **How to organize the data processing ?**



The future – European Chip Design Platform with open-source



- Chips JU initiative
- Cloud based design environment with EDA tools, flows and IP's
- Open source is part of the platform: Analog, RF, AMS, Digital and Photonics related tools, flows
- Open-source design library: open source IP's

DCT

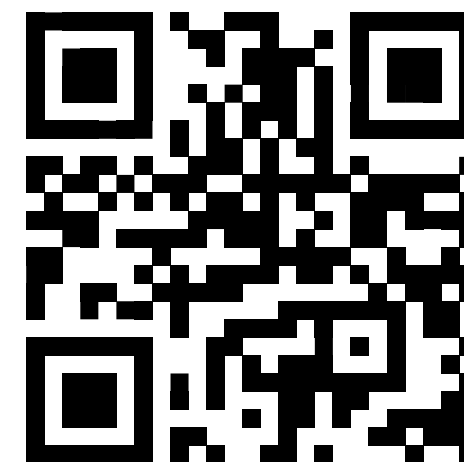
DET

DET

DET



DET call is open right now – ChipFlow will apply so contact Rob Taylor for more details



The future - Low cost MPW



In order to maintain open source MPW IHP plans to have two OpenMPW runs per year

→ SG13CMOS – CMOS only run with AL BEOL where price „P” in EUR is calculated based on total area (mm²) „A” booked

$$P = \begin{cases} - & \text{if } A < 100 \\ 150000/A & \text{if } 100 \leq A \leq 150 \\ 1000 & \text{if } A > 150 \end{cases}$$

→ SG13G2 – fully featured G2 with AL BEOL at approx. 2800 EUR/mm²

The above mentioned prices refer to the open-source designs, where all the views are compatible with the open source EDA tools.

For customers who do not wish to disclose their IP, we offer participation in the OpenMPW program at a 20% discount off the regular price, as the wafers can be shared with other customers.

The turn around processing time is approx. 6 months for CMOS - 8 months for BiCMOS

The typical offer includes 20 bare die samples. Packaging will be offered on request (additional fee can be applied)

The first run can take place 24-th Nov 2025 (CMOS)

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

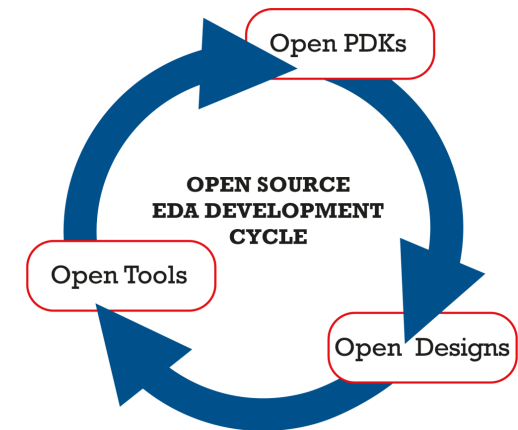
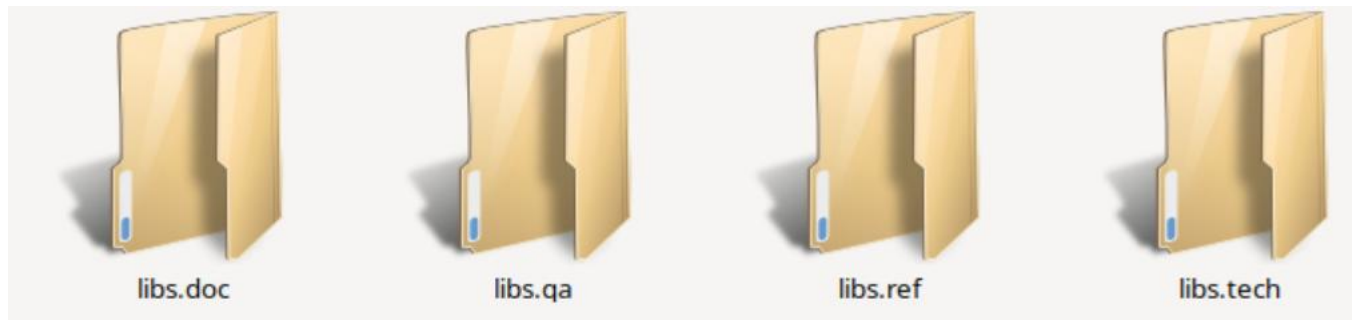
How do we make the PDK ? ?

How do we make the IHP-Open-PDK ?

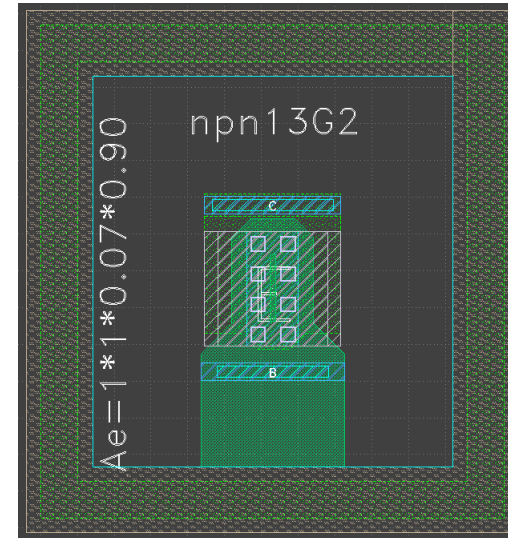
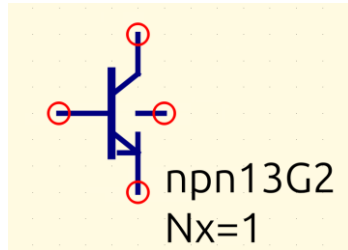
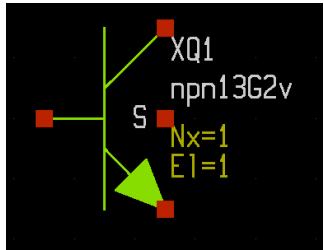


Key development rules of IHP-Open-PDK:

- 0 Compatibility with technology – PDK is defined by technology
- 0 Compatibility with the open-source tools (standardization, dependencies)
- 0 Tight link between Cadence SG13G2 PDK and its open-source version
- 0 Four+ eyes principle – review process



Device integration process



Check list:

- 0 Xschem symbol
- 0 Qucs-S symbol
- 0 Ngspice model
- 0 Xyce model
- 0 Verilog-A model
- 0 Xschem example
- 0 Qucs-S example
- 0 Klayout PyCell
- 0 Magic Pcell
- 0 DRC -test
- 0 LVS extraction rules
- 0 LVS - test

- 0 \$PDK_ROOT/\$PDK/libs.tech/xschem/sg13g2_pr/npn13G2.sym
- 0 \$PDK_ROOT/\$PDK/libs.tech/xschem/sg13g2_tests/
- 0 \$PDK_ROOT/\$PDK/libs.tech/qucs-s/user_lib/IHP_nonlinear...*.lib
- 0 \$PDK_ROOT/\$PDK/libs.tech/qucs-s/examples/
- 0 \$PDK_ROOT/\$PDK/libs.tech/ngspice/model/cornerHBT.lib
- 0 \$PDK_ROOT/\$PDK/libs.tech/xyce/model/cornerHBT.lib
- 0 \$PDK_ROOT/\$PDK/libs.tech/verilog-a/vbic/vacode/vbic_1p3.va
- 0 \$PDK_ROOT/\$PDK/libs.tech/klayout/python/sg13g2_pycell_lib/ihp
- 0 \$PDK_ROOT/\$PDK/libs.tech/magic/ *ongoing
- 0 \$PDK_ROOT/\$PDK/libs.tech/klayout/tech/drc
- 0 \$PDK_ROOT/\$PDK/libs.tech/klayout/tech/lvs

Device model development



Current modelling approach:

- 0 Based on Spice
- 0 Partially supported by Verilog-A models via OSDI interface

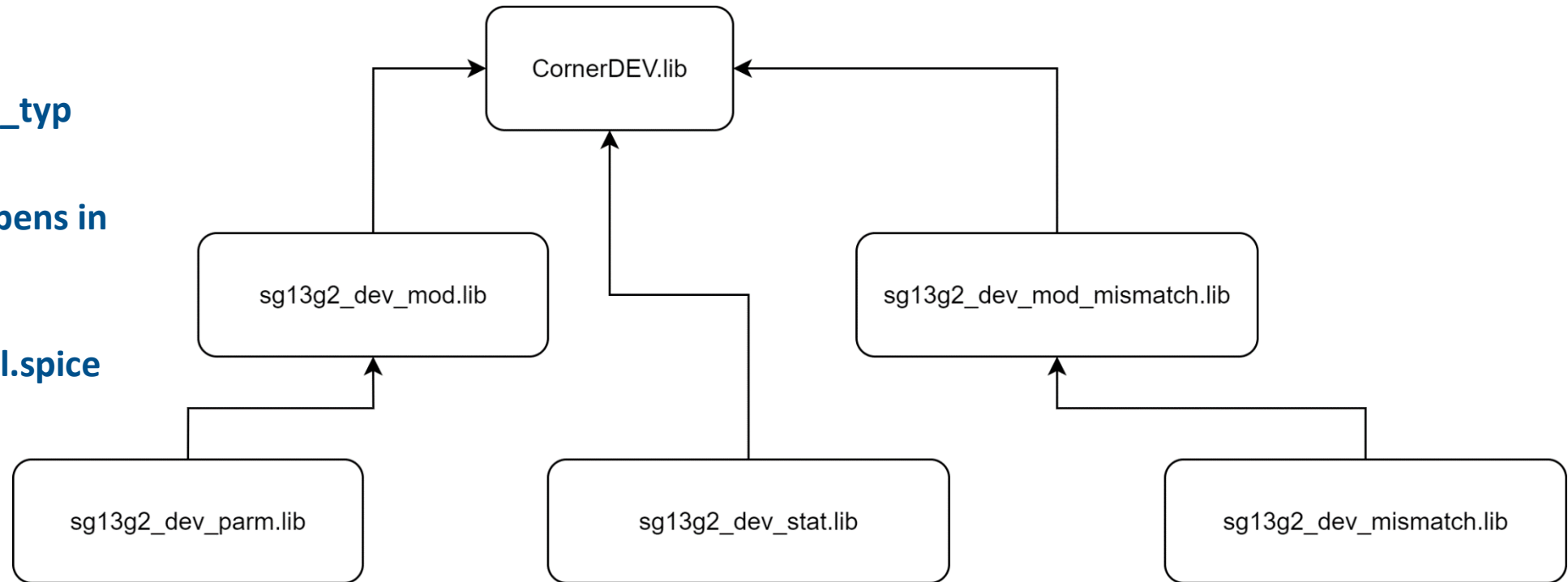
Corners:

- 0 dev_typ, dev_bcs, dev_wcs, dev_ttyp_stat
- 0 mos_tt, mos_ff, mos_ss, mos_fs, mos_sf, mos_tt_stat

.lib CornerDEV.lib dev_typ

**The actual magic happens in
\$HOME/.spiceinit**

.include sg13g2_stdcell.spice



Device model development

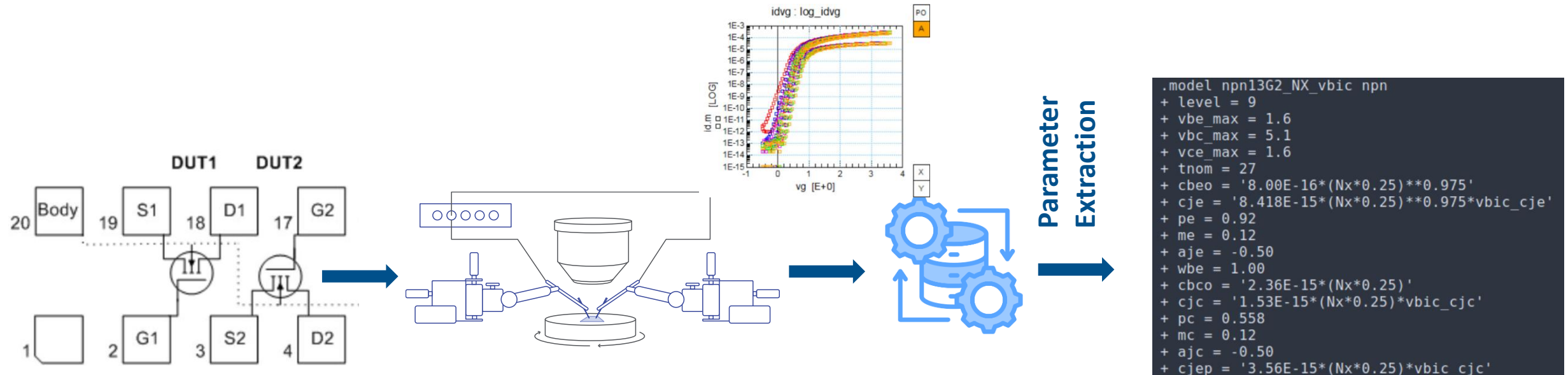


Current modelling approach:

- Industrial flow uses Keysight ICCAP
- Measurements are taken for different conditions: temperature
- Multiple measurements -> statistics (process, global variation)
- Co-simulation spectre, hspice, ngspice

Alternative open-source flow:

- Device Modeling Toolkit by Semimod
- co-simulation with ngspice
- Verilog-A modelcards support



Standard cells development

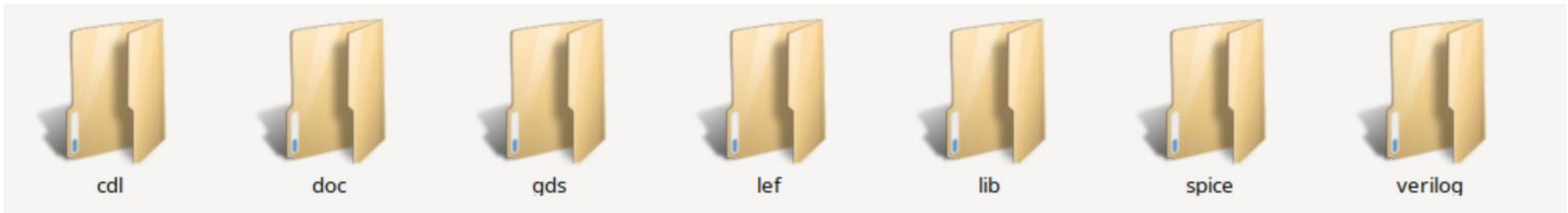


Standard cells reverse engineering:

- 0 GDS -> CDL
- 0 CDL->SPICE
- 0 SPICE->LIB
- 0 CDL->Verilog
- 0 GDS->LEF

Stdcell alternative open-source flow:

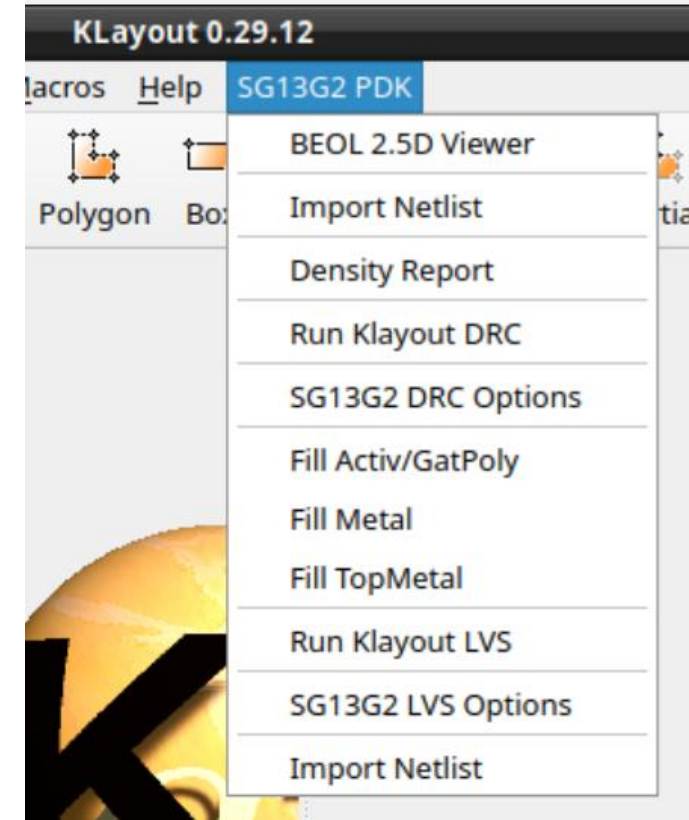
- 0 Verilog
- 0 Schematic/Spice netlist
- 0 SPICE->LIB (Icetime, libretto, CharLib)\
- 0 SPICE->GDS Iclayout
- 0 GDS->LEF (Magic)



Physical verification is critical



- 0 Klayout DRC rule deck is written in RUBY
- 0 \$PDK_ROOT/\$PDK/libs.tech/klayout/tech/drc
- 0 be careful when running DRC during submission it is only a foundry precheck
- 0 Klayout LVS rule deck is written in RUBY
- 0 \$PDK_ROOT/\$PDK/libs.tech/klayout/tech/lvs
- 0 performance – populate jobs on multicore up to 24
- 0 be careful with RAM

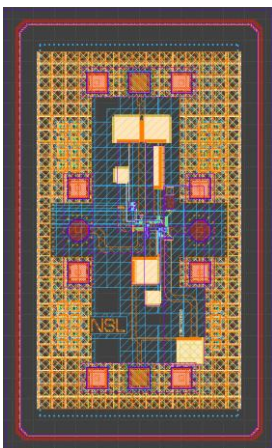


Open-source design library

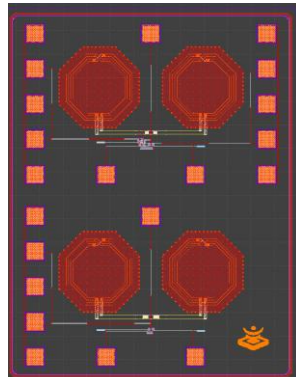
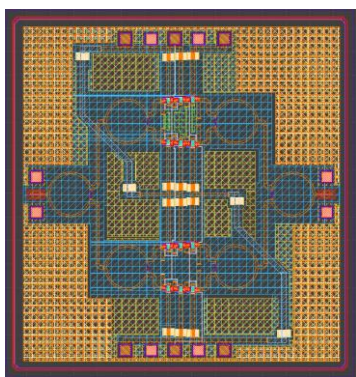
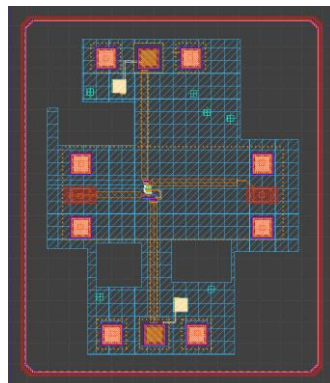
In principle the Design Library does not exist yet but...



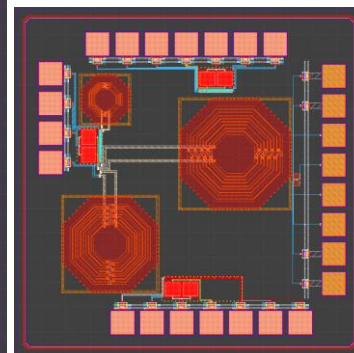
97 GHz TIA



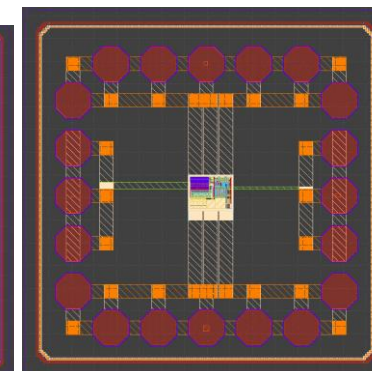
DC to 130 GHz TIA 4-way 180 GHz PA 5GHz Mixer



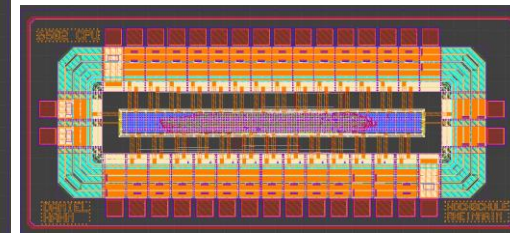
GPS LNA



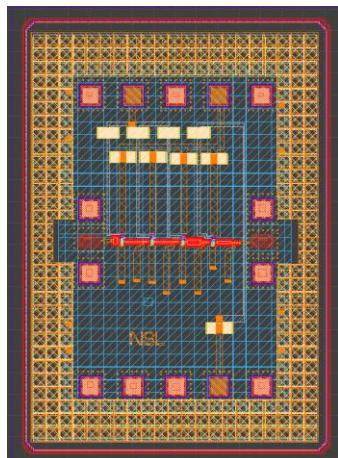
BG - Vref



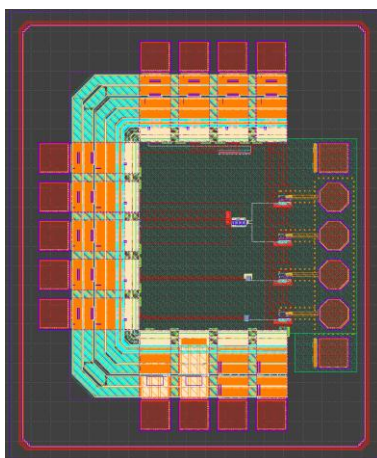
6502 cpu



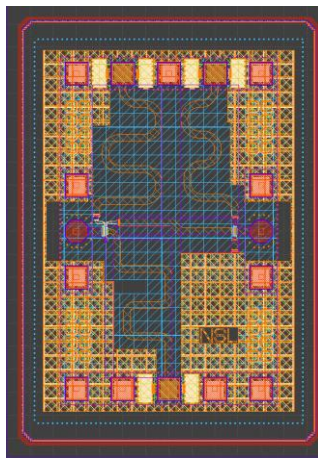
160 GHz LNA



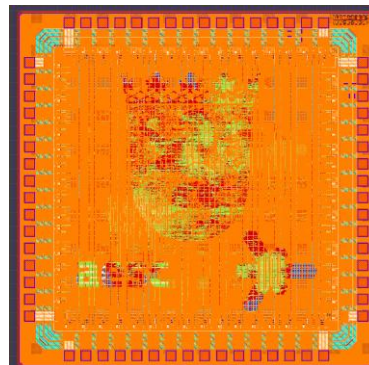
Active L VCO



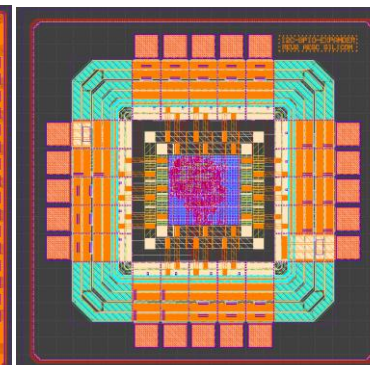
40 GHz LN TIA



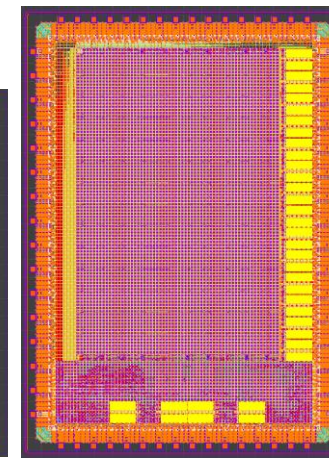
ElemRV-N



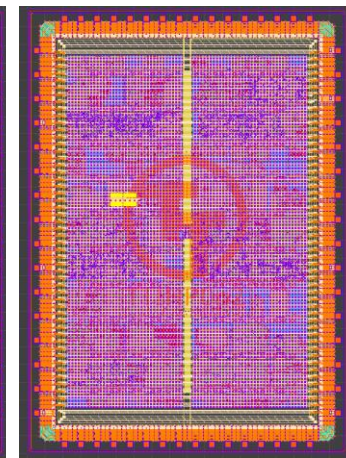
I2c-gpio-expander



Greyhound



2x TT



How to reach us



Email

→ openpdk@ihp-microelectronics.com

GitHub

→ issues – for reporting issues, questions

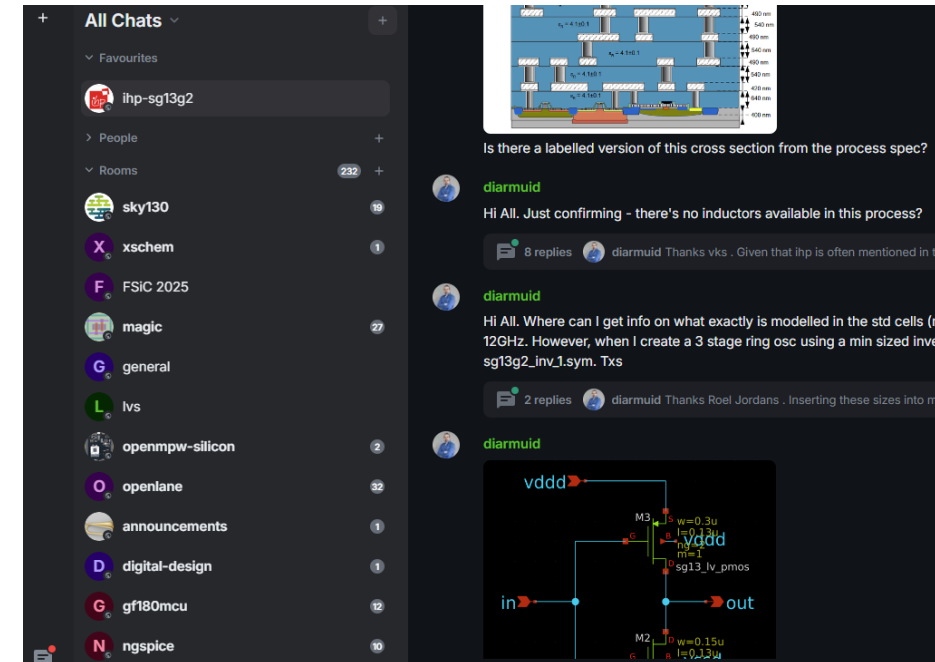
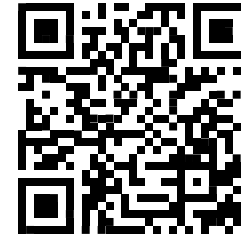
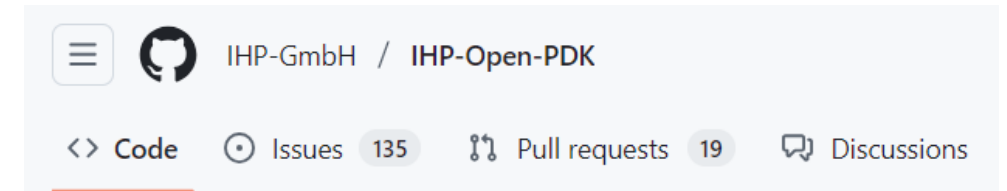
→ discussions – for requesting features

Open Source Fossi-Chat.org channel:

#ihp-sg13g2

→ general discussions

→ announcements



Funding opportunities



<https://nl.net.nl/>

[Amaranth HDL](#)

Design FPGAs and ASICs in Python

[Apicula IO primitives](#)

Add additional IO primitives to libre Gowin FPGA tools

[FABulous Demo SoC](#)

SoC with open source FPGA based on FABulous

[FPGA-ISP-UVC-USB2](#)

Open hardware FPGA-based USB webcam

[LiteX](#)

Developer framework for FPGA and ASIC designs

[LibreSilicon: Pad Cell Generator](#)

Custom pad cells for integrated chip layout generation

[Test Procedures for MOSFET SPICE Model Validation](#)

[pcb-rnd, sch-rnd](#)

Open source EDA suite

[Send in your ideas. Deadline October 1, 2025](#)

To our team at IHP



- open positions
- student's internship

To all members of the open-source community who have supported and contributed our initiatives



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

