

Chapter 5 - Process Design Kit (PDK)

Course authors (Git file)



- 1 What is a PDK?
- 2 Open-Source PDK and GitHub
- 3 Content of the PDK ihp-sg13g2
- 4 File formats
- 5 Example: Cell AO21
- 6 Ruleset documents



Section 1

What is a PDK?



Wikipedia definition

A process design kit (PDK) is a set of files used within the semiconductor industry to model a fabrication process for the design tools used to design an integrated circuit. The PDK is created by the foundry defining a certain technology variation for their processes. ...

... The designers use the PDK to design, simulate, draw and verify the design before handing the design back to the foundry to produce chips. The data in the PDK is specific to the foundry's process variation and is chosen early in the design process, influenced by the market requirements for the chip. An accurate PDK will increase the chances of first-pass successful silicon.

Source: https://en.wikipedia.org/wiki/Process_design_kit



Open-source viewpoint

- Semiconductor industry started to integrate open-source.
- Open-source PDKs created by Semiconductor Fabs were a “missing link” between:
 - Open-source EDA tools (RTL-to-GDS) and
 - Microchip production (GDS-to-Chip)
- Since there is open-source PDKs, the growth of the open-source ecosystem is measurable.
- Many of the tools have been the classical “one-person maintained” open-source projects. It is getting better.



In the context of this course

The PDK (ihp-sg13g2) integrates seamless (to the user) into the OpenROAD flow scripts toolchain.

We have seen reference points from the tools onto the PDK in:

- the configuration files
- the structure of the design directories
- some Variables



Naming

PDKs sometimes are referred to as:

- Process design kits
- Process node
- Technology node
- Technology



Section 2

Open-Source PDK and GitHub



Difference from closed source

With publishing a PDK under a open-source license, the development from there on becomes a worldwide visible joint effort. The number of contributors and authors of the PDK can only increase from here on.



Collaborative workflow in GitHub

Some of the main principles of open-source are the permissions to use, study, change and re-distribute the published code and data according to the license. This leads to a open collaboration in which everyone can participate.

GitHub enables a workflow that was designed and build with these principles and opportunities in mind. A good starting point to explore the open collaboration in the IHP PDK are

- Issues (open and closed)
- Pull requests (open and closed)

The topics and discussions that you can read and study there will draw a picture of how the process of open collaboration works for the PDK.



Issues open

🔥 **Want to contribute to IHP-GmbH/IHP-Open-PDK?** Dismiss ▾

If you have a bug or an idea, read the [contributing guidelines](#) before opening an issue.

Filters ▾

🏷 Labels 9
📅 Milestones 0
New issue


✕ Clear current search query, filters, and sorts

🕒 35 Open ✓ 55 Closed	Author ▾	Label ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
🕒 [bug] ifnone state-dependent path delay #209 opened 2 weeks ago by likeamahoney					👤	💬 5
🕒 [bug] LRM specify block delay path restrictions bug #208 opened 2 weeks ago by likeamahoney						💬 3
🕒 Simulation of MOSFET noise with ngspice #207 opened 2 weeks ago by 0y8w1x						💬 2
🕒 .spiceinit seems to produce errors in AC simulation bug #205 opened 3 weeks ago by olisnr					👤	💬 2
🕒 how to use tolerances in LVS? #203 opened 3 weeks ago by olisnr						💬 5
🕒 DRC seem to miss error on GatPoly Gat.b or Gat.b1 bug #201 opened 3 weeks ago by olisnr						💬 24

Figure 1: Issues open



Issues closed

 **Want to contribute to IHP-GmbH/IHP-Open-PDK?**
Dismiss ▾

If you have a bug or an idea, read the [contributing guidelines](#) before opening an issue.

Filters ▾
Labels 9
Milestones 0
New Issue

✕ Clear current search query, filters, and sorts


<input type="radio"/> 35 Open <input checked="" type="radio"/> 55 Closed	Author ▾	Label ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
<input checked="" type="radio"/> .spice net for LVS #197 by olisnr was closed 3 weeks ago						11
<input checked="" type="radio"/> a question to LVS from .spice #196 by olisnr was closed last month						2
<input checked="" type="radio"/> Resistor parameter <code>b</code> usage? question #191 by hpretl was closed 3 weeks ago						5
<input checked="" type="radio"/> sram: missing .lib files #177 by dnltz was closed on Aug 12						5
<input checked="" type="radio"/> Any Recomend Python Versions? question #169 by redpanda3 was closed 3 weeks ago						12
<input checked="" type="radio"/> Routing issue in <code>dfrbp_1</code> flop bug invalid #166 by sergelandreyev was closed on Aug 5						2

Figure 2: Issues clsoed



Pull requests closed

First time contributing to IHP-GmbH/IHP-Open-PDK?

If you know how to fix an [issue](#), consider opening a pull request for it. You can read this repository's [contributing guidelines](#) to learn how to open a good pull request.

[Dismiss](#) ...

Filters ▾

Q is:pr is:closed

Labels 9

Milestones 0

New pull request

✕ Clear current search query, filters, and sorts

3 Open ✓ 115 Closed	Author ▾	Label ▾	Projects ▾	Milestones ▾	Reviews ▾	Assignee ▾	Sort ▾
iho-sg13g2: libs.ref: sg13g2_io: verilog: Fix specify syntax ✓ #215 by dnlitz was merged last week							1
Update KLayout DRC scripts ✓ #214 by akrinke was merged last week							
PyCell Klayout integration ✓ #213 by ThomasZecha was merged last week							
LVS rule decks: Fix GF180 remnants in log strings ✗ #212 by martinjankoehler was merged last week							3
Updated procedure to avoid constant opening cmd window on Windows to... ✓ #211 by adatsuk was merged last week 2 tasks							2

Figure 3: Pull requests closed



License file

The IHP open-source PDK is published with an Apache 2.0 license:

<https://github.com/IHP-GmbH/IHP-Open-PDK/blob/main/LICENSE>

Apache 2.0 is a permissive open-source license. Read more about different open-source licenses here:

<https://choosealicense.com/licenses/>

Tip:

Know about the permissions, conditions and limitations of the licenses you are using for your projects!



Section 3

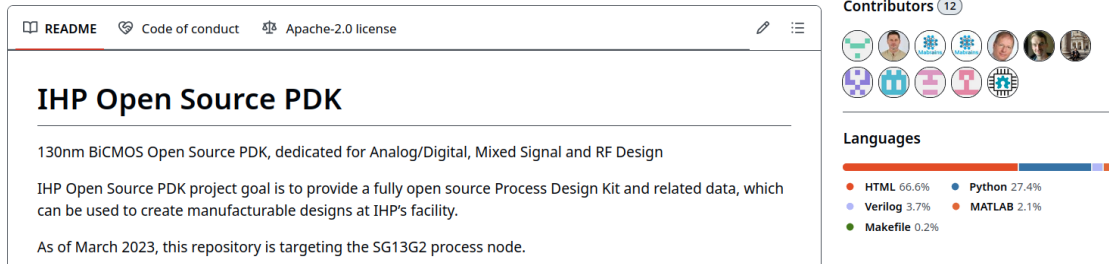
Content of the PDK ihp-sg13g2



The README

The Readme file in the PDKs repository is the starting point for information about the content of the PDK.

<https://github.com/IHP-GmbH/IHP-Open-PDK/blob/main/README.md>



IHP Open Source PDK

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

IHP Open Source PDK project goal is to provide a fully open source Process Design Kit and related data, which can be used to create manufacturable designs at IHP's facility.

As of March 2023, this repository is targeting the SG13G2 process node.

Contributors 12

Languages

- HTML 66.6%
- Python 27.4%
- Verilog 3.7%
- MATLAB 2.1%
- Makefile 0.2%

Figure 4: Readme



Project roadmap

A GANTT chart of the roadmap for the open-source PDK is available under this weblink. It shows the projects timeline (2022 - 2026):

https://github.com/IHP-GmbH/IHP-Open-PDK/blob/main/ihp-sg13g2/libs.doc/roadmap/open_PDK_gantt.png

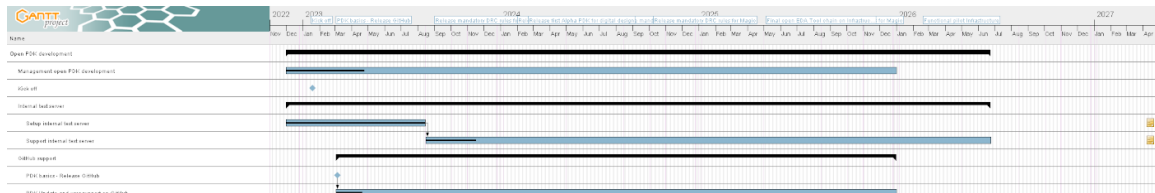


Figure 5: Gantt chart



Tool support

The PDK supports a bunch of tools with the included file formats. A list of the file types and formats follows later in this chapter.

The main tools for this course are

- OpenROAD flow scripts
- OpenROAD
- KLayout



The list of supported tools from the README in the PDK:

Supported EDA Tools

- ngspice
 - Download: <https://ngspice.sourceforge.io/download.html>
 - Source: <https://sourceforge.net/p/ngspice/ngspice/ci/master/tree>
- Xyce
 - Download: <https://xyce.sandia.gov/downloads/executables>
 - Source: <https://github.com/Xyce/Xyce>
- xschem
 - Source: <https://github.com/StefanSchipppers/xschem>
- Qucs-S
 - Download: <https://ra3xdh.github.io>
 - Source: https://github.com/ra3xdh/qucs_s
- KLayout
 - Download: <https://www.klayout.de/build.html>
 - Source: <https://github.com/KLayout/klayout>
- OpenEMS
 - Source: <https://github.com/thliebig/openEMS-Project>
- OpenROAD
 - Source: <https://github.com/The-OpenROAD-Project/OpenROAD>
- OpenROAD-flow-scripts
 - Source: <https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts>

Figure 6: Supported tools



Cells in the PDK

There are four different sets of cells (or devices) in the PDK:

- Base cellset with limited set of standard logic cells
 - CDL, GDSII, LEF, Tech LEF
 - Liberty, SPICE Netlist, Verilog
- IO cellset
 - GDSII, LEF, Liberty (dummy), SPICE Netlist
- SRAM cellset
 - CDL, GDSII, LEF, Liberty, Verilog
- Primitive devices
 - GDSII



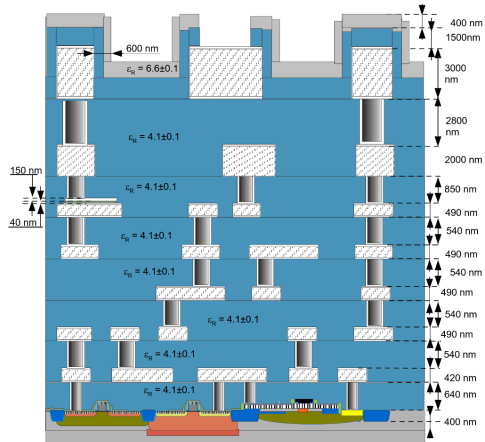
Other data in the PDK

- KLayout tool data:
 - layer property and tech files
 - DRC rules (minimal set)
 - PyCells
 - initial version of the wrapper API
 - sample cells
- Pcells (for reference only) `libs.tech/pycell`
- MOS/HBT/Passive device models for ngspice/Xyce
- xschem: primitive device symbols, settings and testbenches
- OpenEMS: tutorials, scripts, documentation
- SG13G2 Process specification & Layout Rules
- MOS/HBT Measurements in MDM format
- Project Roadmap Gantt chart



Layer stack

IHP sg13g2 Layers in a picture. [Source PDF](#)



Design rules

- The DRC (design rules check) with ORFS happens in KLayout.
- The data for the minimal and maximal checks is here:

<https://github.com/IHP-GmbH/IHP-Open-PDK/tree/main/ihp-sg13g2/libs.tech/klayout/tech/drc>



IHP-Open-PDK / ihp-sg13g2 / libs.tech / klayout / tech / drc /

Add file ▾



akrinke Update KLayout DRC scripts



194ef09 · 5 months ago

History

Name	Last commit message	Last commit date
..		
MissingRules_maximal.md	Update KLayout DRC scripts	8 months ago
README.md	Update KLayout DRC scripts	8 months ago
README_maximal.md	Update KLayout DRC scripts	8 months ago
README_minimal.md	KLayout DRC: added first version of full DRC rule deck	9 months ago
sg13g2_maximal.lydrc	Update KLayout DRC scripts	5 months ago
sg13g2_minimal.lydrc	Update KLayout DRC scripts	5 months ago

README.md

Minimum Rule Set - [README](#)Maximum Rule Set - [README](#), [MissingRules](#)

Figure 8: DRC files lydrc for KLayout



Layout versus Schematic

- The LVS check with ORFS happens in KLayout.
- The data for the LVS is here:

<https://github.com/IHP-GmbH/IHP-Open-PDK/tree/main/ihp-sg13g2/libs.tech/klayout/tech/lvs>



IHP-Open-PDK / ihp-sg13g2 / libs.tech / klayout / tech / lvs /

Add file



sergeiandreyev Merge pull request #224 from TinyTapeout/pr-fix-180-ref

6a89d4e · 4 months ago

History

Name	Last commit message	Last commit date
..		
images	Updating GUI menus to automatically detect active cell	9 months ago
rule_decks	Merge pull request #212 from martinjankoebler/fix-gf180-remnants	5 months ago
testing	Updating custom writer for LVS runset	7 months ago
README.md	Updating GUI menus to automatically detect active cell	9 months ago
run_lvs.py	Adding some deep/flat tests, Updating FETs/RFFETs derivations, Adding...	9 months ago
sg13g2.lvs	Updating chip derivation, Adding more logs for lvs run, updating lvs ...	9 months ago
sg13g2_full.lylvs	Merge pull request #224 from TinyTapeout/pr-fix-180-ref	4 months ago

Figure 9: LVS files lylvs for KLayout



Section 4

File formats



List of file types and formats

CDL: Circuit design language [Link](#)

LEF: Library Exchange Format [Link](#)

TechLEF: Technology LEF [Link](#)

GDS II: Graphic data system II [Link](#)

lib: Liberty timing and power file [Link](#)

sym: Symbol file (Xschem) [Link](#)

sch: Schematic file (Xschem) [Link](#)

lyp: Layer properties file (KLayout)

lyt: Technology mapping file (KLayout)

lydrc: DRC rules file (KLayout)

lylvs: LVS rule deck (KLayout)



Section 5

Example: Cell AO21



Cell AO21: GDS in KLayout

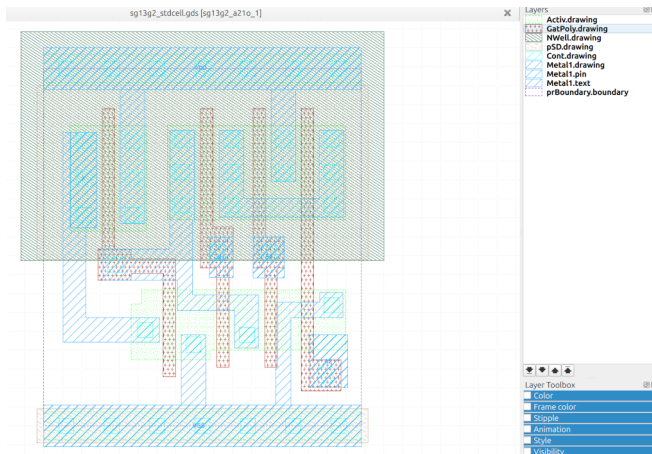


Figure 10: Cell AO21 GDS



Cell AO21: VERILOG HDL language

```
1 // type: AO21
2 `timescale 1ns/10ps
3 `celldefine
4 module sg13g2_a21o_1 (X, A1, A2, B1);
5     output X;
6     input A1, A2, B1;
7
8     // Function
9     wire int_fwire_0;
10
11     and (int_fwire_0, A1, A2);
12     or (X, int_fwire_0, B1);
13
14     // Timing
15     specify
16         (A1 => X) = 0;
17         (A2 => X) = 0;
18         if (A1 == 1'b1 & A2 == 1'b0)
19             (B1 => X) = 0;
20         if (A1 == 1'b0 & A2 == 1'b1)
21             (B1 => X) = 0;
22         ifnone (B1 => X) = 0;
23     endspecify
24 endmodule
25 `endcelldefine
```

Cell AO21: SPICE Netlist

```

1  * Library name: sg13g2_stdcell
2  * Cell name: sg13g2_a21o_1
3  * View name: schematic
4  * Inherited view list: spectre cmos_sch cmos.sch schematic veriloga ahdl
5  * pspace dspf
6  .subckt sg13g2_a21o_1 A1 A2 B1 VDD VSS X
7  XN0 net1 A1 net2 VSS sg13_lv_nmos w=640.00n l=130.00n ng=1 ad=0 as=0 pd=0 ps=0 m=1
8  XN1 net2 A2 VSS VSS sg13_lv_nmos w=640.00n l=130.00n ng=1 ad=0 as=0 pd=0 ps=0 m=1
9  XN2 net1 B1 VSS VSS sg13_lv_nmos w=640.00n l=130.00n ng=1 ad=0 as=0 pd=0 ps=0 m=1
10 XN3 X net1 VSS VSS sg13_lv_nmos w=740.00n l=130.00n ng=1 ad=0 as=0 pd=0 ps=0 m=1
11 XP0 net1 B1 net3 VDD sg13_lv_pmos w=1.000u l=130.00n ng=1 ad=0 as=0 pd=0 ps=0 m=1
12 XP1 net3 A1 VDD VDD sg13_lv_pmos w=1.000u l=130.00n ng=1 ad=0 as=0 pd=0 ps=0 m=1
13 XP2 net3 A2 VDD VDD sg13_lv_pmos w=1.000u l=130.00n ng=1 ad=0 as=0 pd=0 ps=0 m=1
14 XP3 X net1 VDD VDD sg13_lv_pmos w=1.12u l=130.00n ng=1 ad=0 as=0 pd=0 ps=0 m=1
15 .ends
16 * End of subcircuit definition.

```



Cell AO21: Circuit design language

```

1 *****
2 * Library Name: sg13g2_stdcell
3 * Cell Name:    sg13g2_a21o_1
4 * View Name:    schematic
5 *****
6
7 .SUBCKT sg13g2_a21o_1 A1 A2 B1 VDD VSS X
8 *.PININFO A1:I A2:I B1:I X:O VDD:B VSS:B
9 MN0 net1 A1 net2 VSS sg13_lv_nmos m=1 w=640.00n l=130.00n ng=1
10 MN1 net2 A2 VSS VSS sg13_lv_nmos m=1 w=640.00n l=130.00n ng=1
11 MN2 net1 B1 VSS VSS sg13_lv_nmos m=1 w=640.00n l=130.00n ng=1
12 MN3 X net1 VSS VSS sg13_lv_nmos m=1 w=740.00n l=130.00n ng=1
13 MP0 net1 B1 net3 VDD sg13_lv_pmos m=1 w=1.000u l=130.00n ng=1
14 MP1 net3 A1 VDD VDD sg13_lv_pmos m=1 w=1.000u l=130.00n ng=1
15 MP2 net3 A2 VDD VDD sg13_lv_pmos m=1 w=1.000u l=130.00n ng=1
16 MP3 X net1 VDD VDD sg13_lv_pmos m=1 w=1.12u l=130.00n ng=1
17 .ENDS

```



Cell AO21: LEF

```

1  MACRO sg13g2_a21o_1
2      CLASS CORE ;
3      ORIGIN 0 0 ;
4      FOREIGN sg13g2_a21o_1 0 0 ;
5      SIZE 3.36 BY 3.78 ;
6      SYMMETRY X Y ;
7      SITE CoreSite ;
8      PIN A2
9          DIRECTION INPUT ;
10         USE SIGNAL ;
11         ANTENNAMODEL OXIDE1 ;
12         ANTENNAGATEAREA 0.2132 LAYER Metal1 ;
13         PORT
14             LAYER Metal1 ;
15             RECT 2.81 0.405 3.215 0.965 ;
16         END
17     END A2
18     PIN A1
19         DIRECTION INPUT ;
20         USE SIGNAL ;
21         ANTENNAMODEL OXIDE1 ;
22         ANTENNAGATEAREA 0.2132 LAYER Metal1 ;
23         PORT
24             LAYER Metal1 ;
25             RECT 2.215 1.565 2.545 2 ;
26         END
27     END A1

```

Cell AO21: Schematic in XScheme

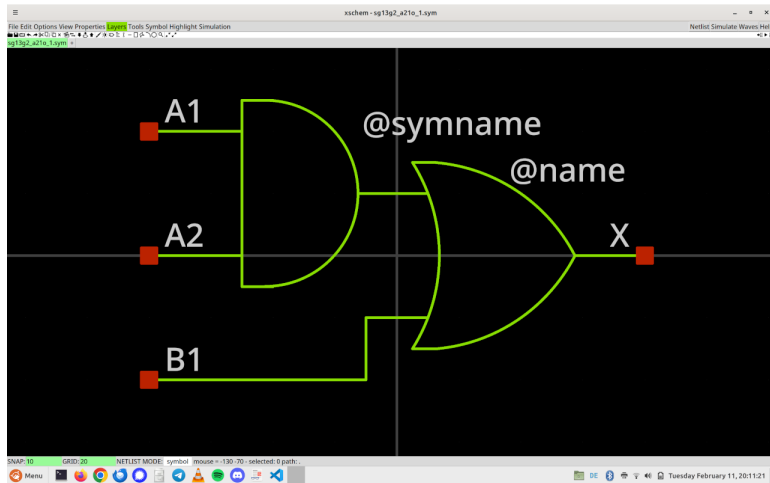


Figure 11: AO21 in XScheme



Liberty files

- Liberty files (.lib) contain information about timing, power and temperature of the cells.
- The ihp130-sg13g2 PDK contains six different Liberty files for the standard cells.
- These six files are categorized by nominal voltages and nominal temperatures:

Liberty file	Voltage	Temperature
sg13g2_stdcell_fast_1p32V_m40C.lib	1.32 V	-40° C
sg13g2_stdcell_fast_1p65V_m40C.lib	1.65 V	-40° C
sg13g2_stdcell_slow_1p08V_125C.lib	1.08 V	125° C
sg13g2_stdcell_slow_1p35V_125C.lib	1.35 V	125° C
sg13g2_stdcell_typ_1p20V_25C.lib	1.20 V	25° C
sg13g2_stdcell_typ_1p50V_25C.lib	1.50 V	25° C

- The PDK contains .lib files for RAM macros and io cells too.



Cell AO21: Liberty description

- The cell AO21 (first cell in the standard cell library list) has 622 lines of data in the lib file.
- The cell description contains:
 - First comes the leakage power for the whole truth table on the inputs.
 - After that each pin gets described with its timing and power characteristics
 - The order of the pins starts with the output. Order: X, A1, A2, B1



Cell AO21: Liberty file extract:

```
1  cell (sg13g2_a21o_1) {
2      area : 12.7008;
3      cell_footprint : "AO21";
4      cell_leakage_power : 158.343;
5      leakage_power () {
6          value : 163.606;
7          when : "!A1&!A2&!B1";
8      }
9
10 ...
11
12     pin (X) {
13         direction : "output";
14         function : "((A1+A2)+B1)";
15
16 ...
17
18     pin (B1) {
19         direction : "input";
20         max_transition : 2.5074;
21
22 ...
```



Section 6

Ruleset documents



Layout rules document

<https://github.com/IHP-GmbH/IHP-Open-PDK/tree/main/ihp-sg13g2/libs.doc/doc>



Figure 12: Layour rules document



Process specification document

<https://github.com/IHP-GmbH/IHP-Open-PDK/tree/main/ihp-sg13g2/libs.doc/doc>

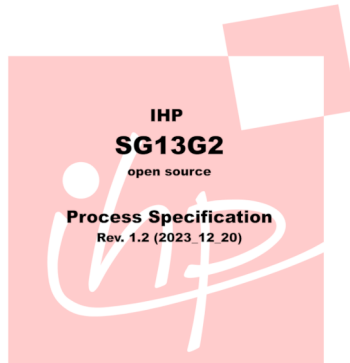


Figure 13: Process specification

