

What is a PDK?

Process Design Kit (PDK) Examination

Thorsten Knoll



- 1 Wikipedia definition
- 2 Open-source viewpoint
- 3 In the context of this course
- 4 Difference from closed source
- 5 Collaborative workflow in GitHub
- 6 Issues openss
- 7 Issues closed
- 8 Pull requests closed
- 9 Resources for you
- 10 Contributing?
- 11 The README
- 12 Project roadmap
- 13 Cells in the PDK



- 14 Other data in the PDK
- 15 Standard cell library
- 16 A single cell from the library
- 17 Layer stack
- 18 Cell AO21: VERILOG HDL language
- 19 Cell AO21: SPICE Netlist
- 20 Cell AO21: Circuit design language



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

ToDo: Image of the border between development and production



In the context of this course

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 openss



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 ▾

🔍 is:open is:issue

🏷 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



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 ▾

🔍 is:issue is:closed

 Labels **9**

 Milestones **0**

New issue

✕ Clear current search query, filters, and sorts

🕒 35 Open ✓ 55 Closed

Author ▾

Label ▾

Projects ▾

Milestones ▾

Assignee ▾

Sort ▾

✓ **.spice net for LVS**

#197 by olisnr was closed 3 weeks ago

💬 11

✓ **a question to LVS from .spice**

#196 by olisnr was closed last month

💬 2

✓ **Resistor parameter **b** usage?** question

#191 by hpretl was closed 3 weeks ago

💬 5

✓ **sram: missing .lib files**

#177 by dnlitz was closed on Aug 12

💬 5

✓ **Any Recommended Python Versions?** question

💬 12

Pull requests closed

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

[Dismiss](#)

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.

 Filters

 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 dnltz 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

Contributing?

Wiki

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>

The screenshot shows the GitHub repository page for 'IHP Open Source PDK'. The main content area contains the title 'IHP Open Source PDK' followed by a description: '130nm BiCMOS Open Source PDK, dedicated for Analog/Digital, Mixed Signal and RF Design'. It then states the project goal: '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.' and mentions the target process node: 'As of March 2023, this repository is targeting the SG13G2 process node.' The right sidebar shows 'Contributors' (12) with a grid of avatars and 'Languages' with a horizontal bar chart and a list of languages and their percentages.

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

Language	Percentage
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

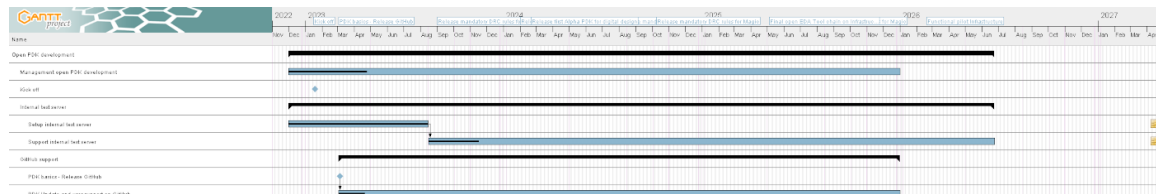


Figure 5: Gantt chart



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



Standard cell library

ToDo:

- Where in the RTI-to-GDS is the cell library needed?
- Design to cells to GDS and Tapeout
- Naming of the cells



A single cell from the library

ToDo:

- Pick one cell
- Pictures of the layers of the cell
 - Klayout
 - 3D Rederings?
- Schematic of the cell



Cell AO21: VERILOG HDL language

```
// type: A021
`timescale 1ns/10ps
`celldefine
module sg13g2_a21o_1 (X, A1, A2, B1);
    output X;
    input A1, A2, B1;

    // Function
    wire int_fwire_0;

    and (int_fwire_0, A1, A2);
    or (X, int_fwire_0, B1);

    // Timing
    specify
```



Cell AO21: SPICE Netlist

```

* Library name: sg13g2_stdcell
* Cell name: sg13g2_a21o_1
* View name: schematic
* Inherited view list: spectre cmos_sch cmos.sch schematic veriloga ahdl
* pspice dspf
.subckt sg13g2_a21o_1 A1 A2 B1 VDD VSS X
XN0 net1 A1 net2 VSS sg13_lv_nmos w=640.00n l=130.00n ng=1 ad=0 as=0 pd=0 ps=0
XN1 net2 A2 VSS VSS sg13_lv_nmos w=640.00n l=130.00n ng=1 ad=0 as=0 pd=0 ps=0
XN2 net1 B1 VSS VSS sg13_lv_nmos w=640.00n l=130.00n ng=1 ad=0 as=0 pd=0 ps=0
XN3 X net1 VSS VSS sg13_lv_nmos w=740.00n l=130.00n ng=1 ad=0 as=0 pd=0 ps=0
XP0 net1 B1 net3 VDD sg13_lv_pmos w=1.000u l=130.00n ng=1 ad=0 as=0 pd=0 ps=0
XP1 net3 A1 VDD VDD sg13_lv_pmos w=1.000u l=130.00n ng=1 ad=0 as=0 pd=0 ps=0
XP2 net3 A2 VDD VDD sg13_lv_pmos w=1.000u l=130.00n ng=1 ad=0 as=0 pd=0 ps=0
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=
.ends

```

Cell AO21: Circuit design language

```

*****
* Library Name: sg13g2_stdcell
* Cell Name:    sg13g2_a21o_1
* View Name:    schematic
*****

.SUBCKT sg13g2_a21o_1 A1 A2 B1 VDD VSS X
*.PININFO A1:I A2:I B1:I X:O VDD:B VSS:B
MN0 net1 A1 net2 VSS sg13_lv_nmos m=1 w=640.00n l=130.00n ng=1
MN1 net2 A2 VSS VSS sg13_lv_nmos m=1 w=640.00n l=130.00n ng=1
MN2 net1 B1 VSS VSS sg13_lv_nmos m=1 w=640.00n l=130.00n ng=1
MN3 X net1 VSS VSS sg13_lv_nmos m=1 w=740.00n l=130.00n ng=1
MP0 net1 B1 net3 VDD sg13_lv_pmos m=1 w=1.000u l=130.00n ng=1
MP1 net3 A1 VDD VDD sg13_lv_pmos m=1 w=1.000u l=130.00n ng=1
MP2 net3 A2 VDD VDD sg13_lv_pmos m=1 w=1.000u l=130.00n ng=1

```

