Leibniz Institute for high performance microelectronics

OpenPDK SG13G2 – Current Status, Roadmap and Open Questions

Sergei Andreev – Scientist Technology / Research & Prototyping Service

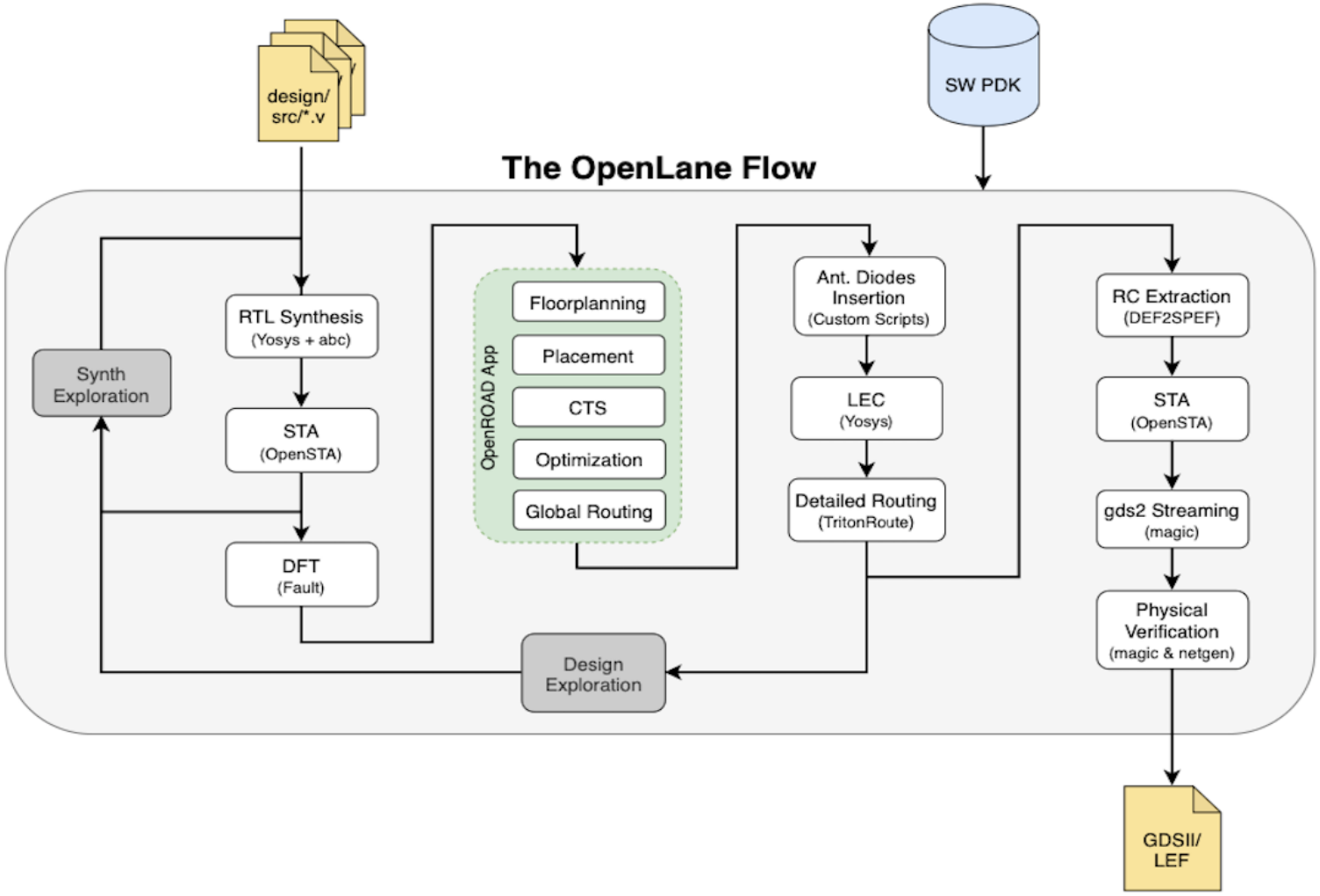
Networking Workshop FMD-QNC ‘OpenPDK, OpenTooling and Open Source Design – An Initiative to Push Development’

June 27/28 2023

OpenPDK and OpenTooling – Status

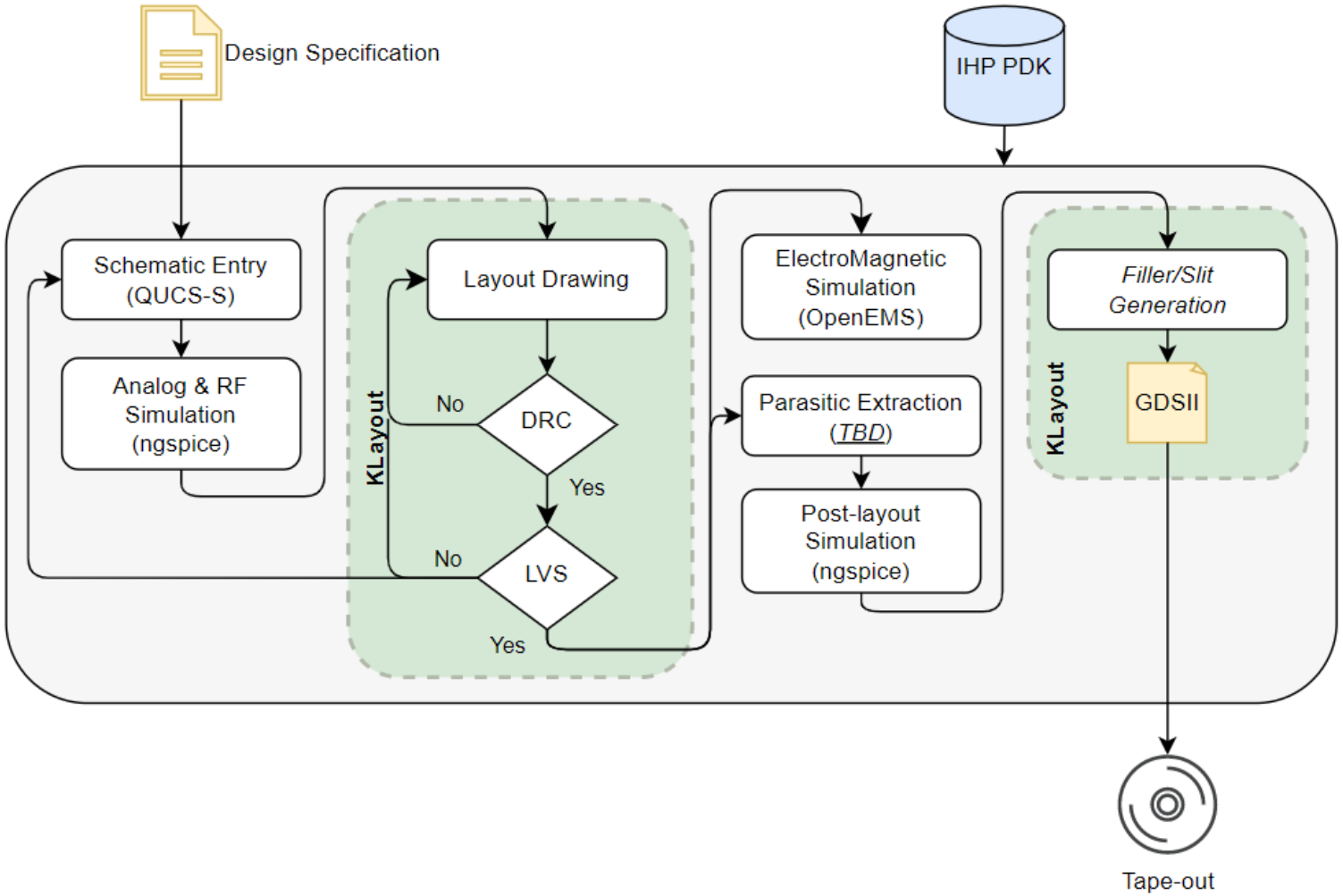
1. IHP Open Source EDA Flows Proposals
2. OpenPDK Project on GitHub
3. Dedicated OpenPDK Virtual Linux Host Machine
4. Available OpenPDK Data
5. Open Questions
6. Next Steps / Planned Updates
7. Summary / Goals

Digital Open Source Development Flow



- Yosys + ABC
- Magic
- Netgen
- CVC
- SPEF-Extractor
- OpenSTA
- KLayout
- Fast/TritonRoute
- TritonCTS
- …

Analog/RF OpenPDK/EDA Flow Proposal

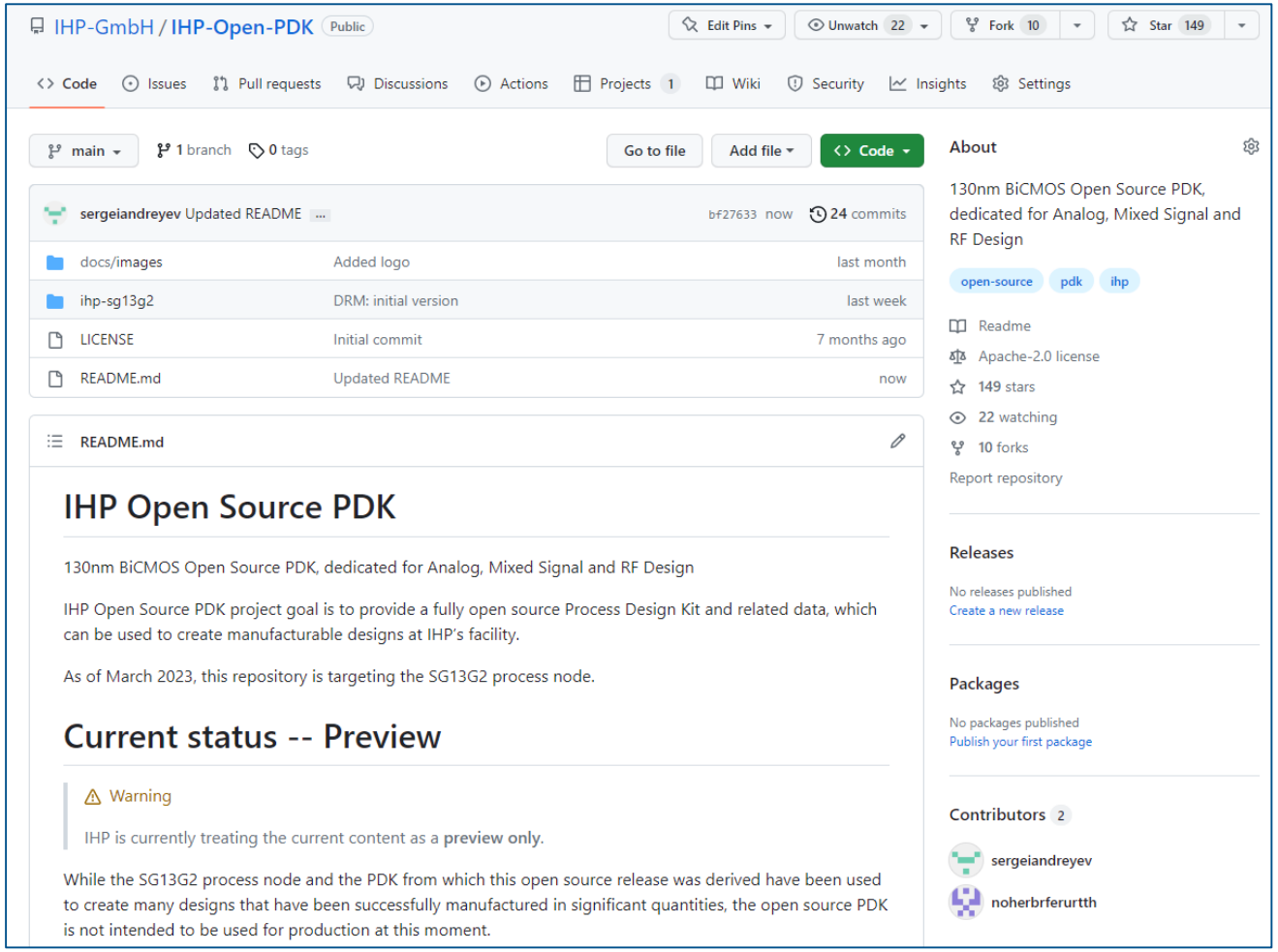


- KLayout-oriented flow
  - Layout design
  - Parameterizable cells
  - Physical Verification
- QUCS-S
- ngspice
- OpenEMS
- … ?

OpenPDK and OpenTooling – Status

1. IHP Open Source EDA Flows Proposals
2. OpenPDK Project on GitHub
3. Dedicated OpenPDK Virtual Linux Host Machine
4. Available OpenPDK Data
5. Open Questions
6. Next Steps / Planned Updates
7. Summary / Goals

OpenPDK Project on GitHub

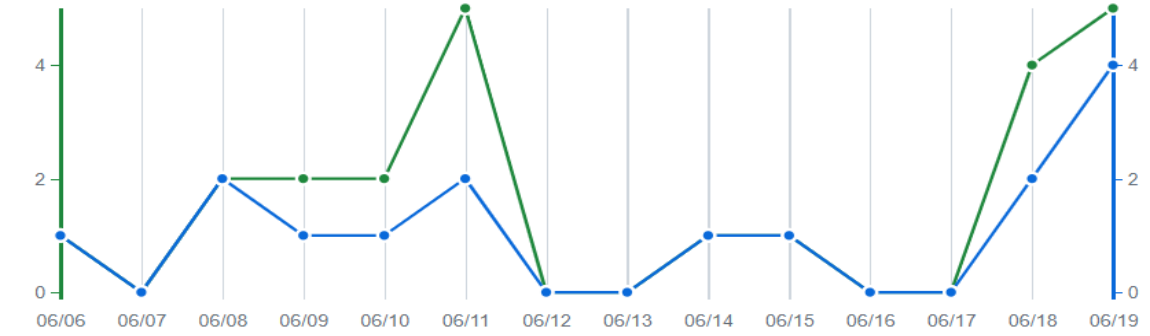


PDK Contents:

- Base cell set with limited set of standard logic cells (Open130-G2)
- GDSII view of primitive devices
- KLayout layer property and tech files
- SPICE Models of HBT devices
- OpenEMS: tutorials, scripts, documentation
- SG13G2 Process specification
- SG13G2 Layout Rules
- MOS/HBT Measurements in MDM format
- Project Roadmap Gantt chart

OpenPDK GitHub Project Traffic (last two weeks)

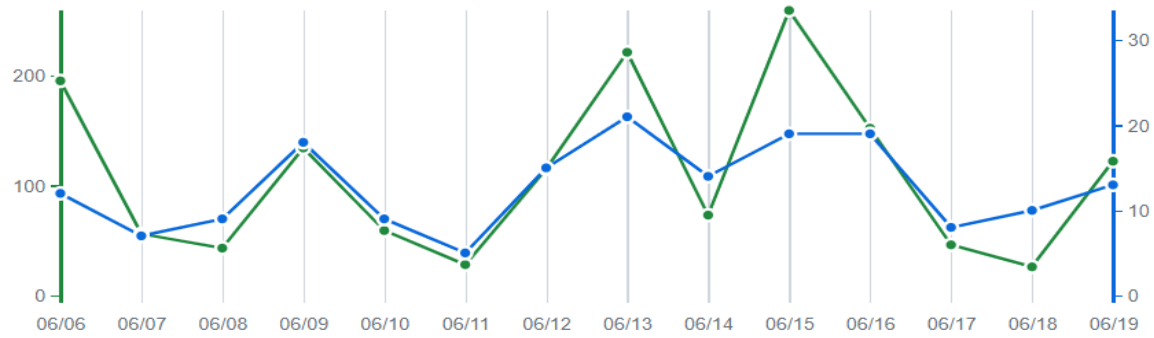
Git clones (June 6 – June 19)



23 Clones

12 Unique cloners

Visitors (June 6 – June 19)



1,529 Views

127 Unique visitors

Referring sites

Site	Views	Unique visitors
Google	507	45
github.com	222	8
t.co	67	8
ngspice.sourceforge.io	25	3
linkedin.com	20	1
DuckDuckGo	8	4
infineon.webex.com	6	1
cn.bing.com	6	1
yandex.ru	1	1
ecosia.org	1	1

OpenPDK and OpenTooling – Status

1. IHP Open Source EDA Flows Proposals
2. OpenPDK Project on GitHub
3. Dedicated OpenPDK Virtual Linux Host Machine
4. Available OpenPDK Data
5. Open Questions
6. Next Steps / Planned Updates
7. Summary / Goals

Dedicated OpenPDK Virtual Linux Host Machine

- Virtual Machine (VMWare VSphere) with Linux OS
  - 4 CPU
  - Memory 16GB
  - HD 1TB
  - OS Ubuntu 22.04.2 LTS
  - SSH, FTP, ThinLinc servers
  - Only internal users by request
  - Automatic backup every 24h at night

PDK	**sky130 [GitHub, open_pdks]**
Layout	**KLayout [Deb package, v0.28]** **Magic [GitHub]** **KiCAD [6.0.2]** **gdspy [1.6.12]** **netgen [GitHub]**
Schematic	**Xschem [GitHub]** **QUCS-S [GitHub]** **Revolution EDA [GitHub]**
Simulation	**Ngspice [GitHub]** **Xyce [GitHub]** **spectre2spice [GitHub]**
Modeling	**DMT [GitHub, user-level]** **OpenVAF [23.2.0]**
EM	**OpenEMS [GitHub]** **Octave [6.4.0]**
Flows	**Open_pdks [GitHub, user-level]** **OpenLane [GitHub, user-level]**
PCells	**OpenPCells [GitHub, user-level]** **Magic TCL & KLayout Python Pcells [GitHub, sky130]**
Gen	**PDKMaster [0.9.0, PyPi]**
Documentation	**doxygen [1.9.1]** **graphviz [2.43.0]**

OpenPDK and OpenTooling – Status

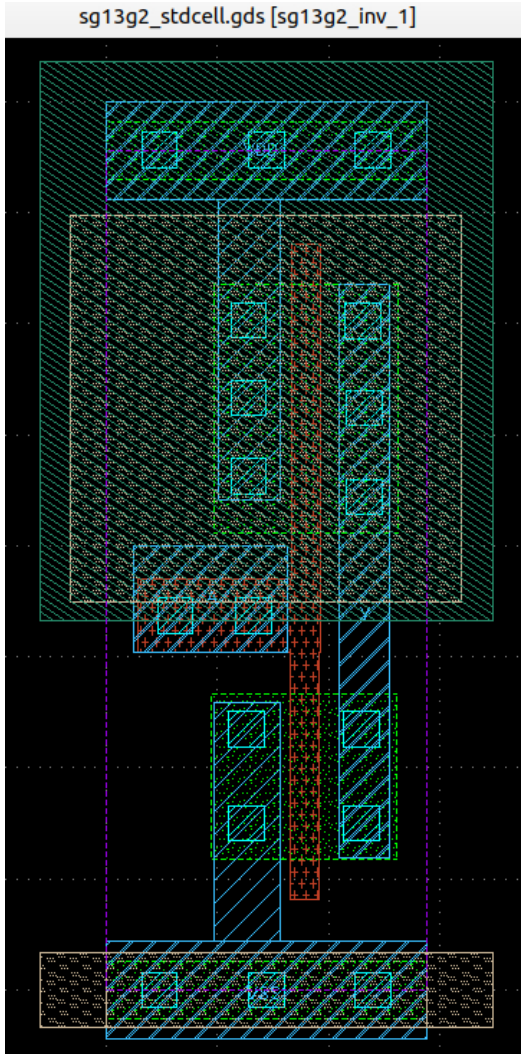
1. IHP Open Source EDA Flows Proposals
2. OpenPDK Project on GitHub
3. Dedicated OpenPDK Virtual Linux Host Machine
4. Available OpenPDK Data
5. Open Questions
6. Next Steps / Planned Updates
7. Summary / Goals

Base Cell Set with Digital Standard Cells

- 78 cells
- Views
  - CDL Netlist
  - GDSII
  - LEF, Tech LEF
  - SPICE Netlist
  - Liberty
  - Verilog
  - OA Library *
- Functions
  - ao[i], [n]and, buf, decap, flops, dly, fillers, inv, mux, [n]or, tie, x[n]or, auxiliary

```
********************************************************************************
* Library Name: sg13g2_stdcell
* Cell Name:    sg13g2_inv_1
* View Name:    schematic
********************************************************************************

.SUBCKT sg13g2_inv_1 A VDD VSS Y
*.PININFO A:I VDD:B VSS:B Y:O
MX1 VSS A Y VSS sg13_lv_nmos m=1 w=740.00n l=130.00n ng=1
MX0 VDD A Y VDD sg13_lv_pmos m=1 w=1.12u l=130.00n ng=1
.ENDS
```

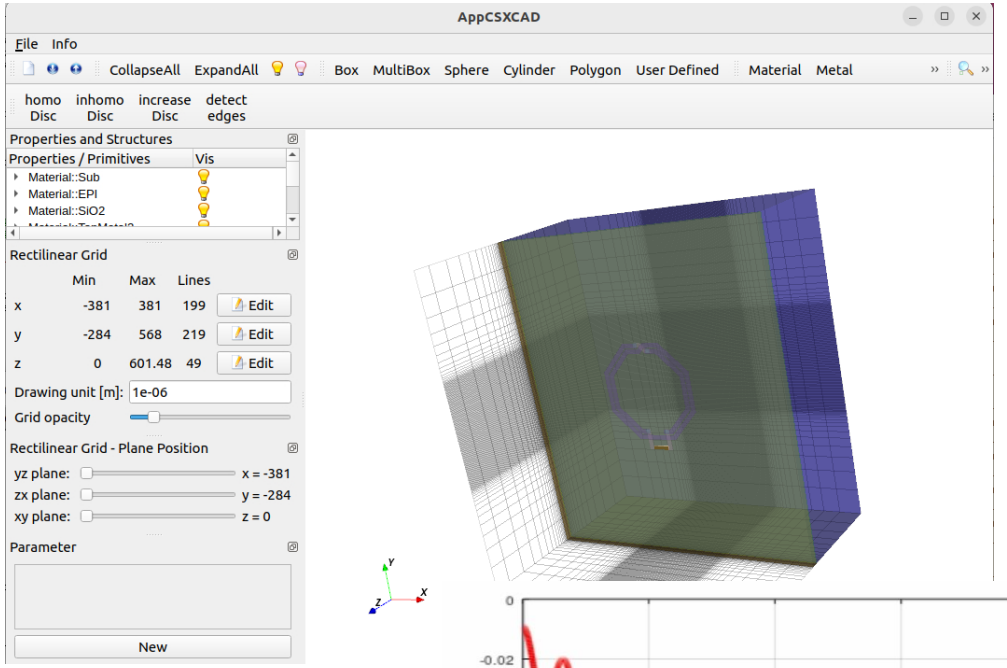
```
// type: IN
`timescale 1ns/10ps
`celldefine
module sg13g2_inv_1 (Y, A);
  output Y;
  input A;

  // Function
  not (Y, A);

  // Timing
  specify
    (A => Y) = 0;
  endspecify
endmodule
`endcelldefine
```

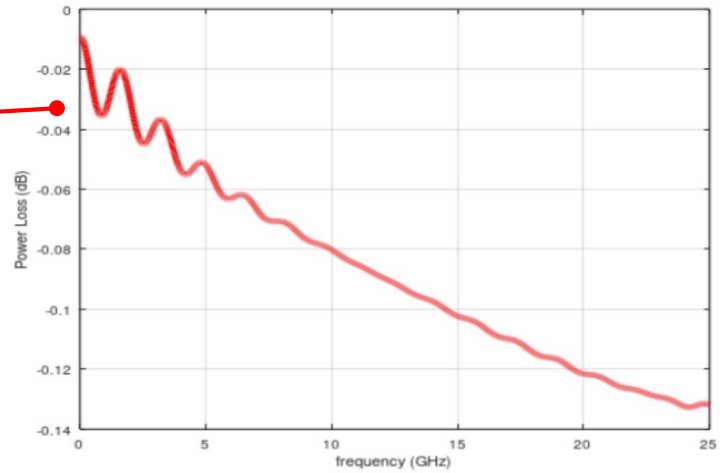


*** not part of the OpenPDK delivery**

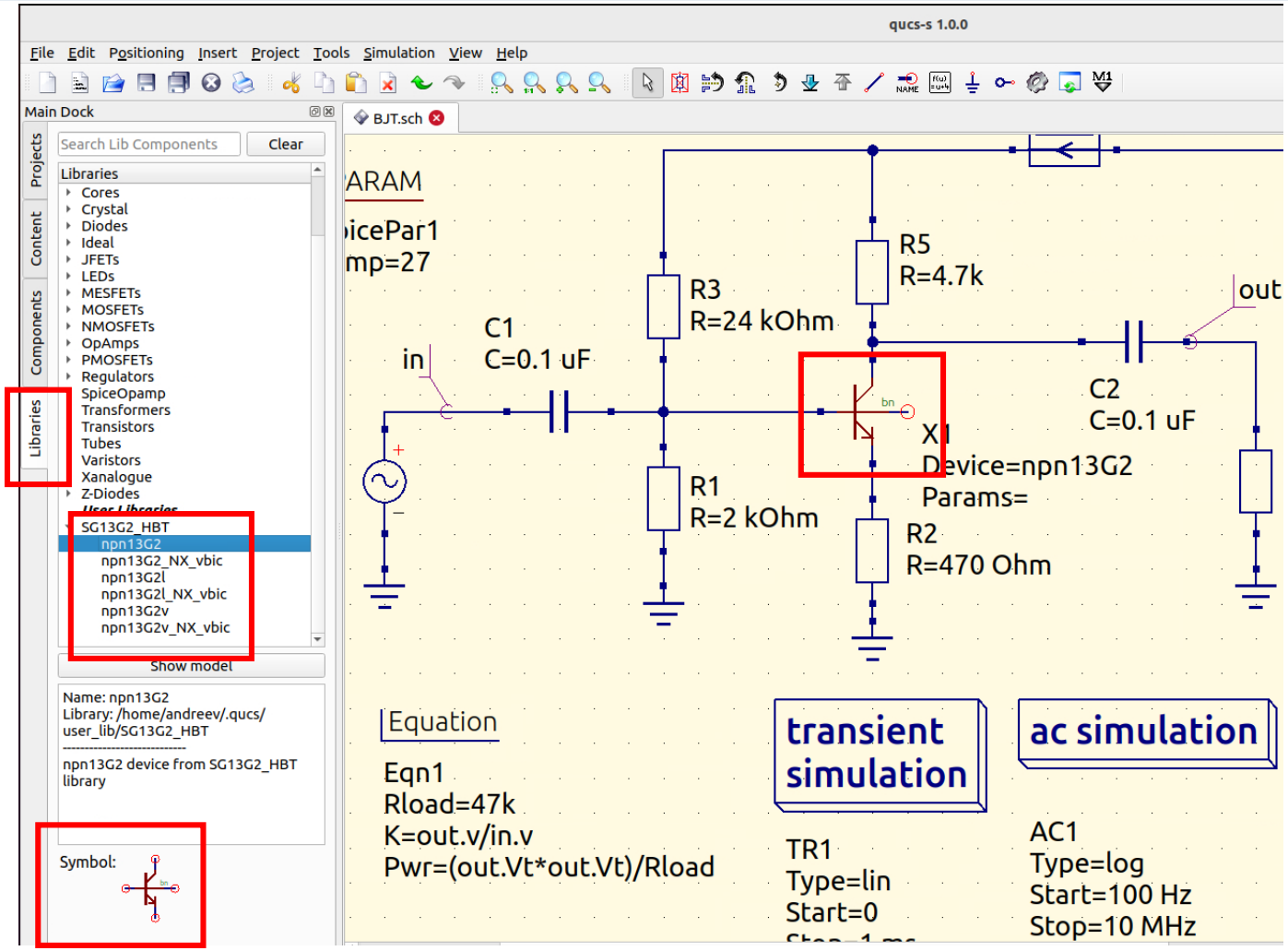
OpenEMS ElectroMagnetic Solver Review

- 3D FDTD solution targeting RF EM simulations
- Model built by Python or Octave/Matlab scripting
- Graphical viewer for model + mesh (CSXCAD)
- Some interfaces to EDA packages, but no KLayout support yet
- No internal support for GDSII import, interface was created using Python library gdspy
- Variable FDTD grid spacing, mesh smoothing
- Semiconductor substrate support (permittivity + conductivity)
- S-Parameter output
- Useful tutorials for RF examples
- **Possible issue: small residual energy at low frequency or DC might create DC leakage in simulation results**
  - Efficient solution in commercial FDTD tools is "resonance estimation“, but that is not implemented in OpenEMS so far
- **Mostly manual mesh definition**
  - automatic meshing add-on project (OpenMesh) unfinished





QUCS-S Custom Library with IHP OpenPDK Devices




```
<Symbol>
  <Line -30 0 10 0 #000080 2 1>
  <Line -20 0 10 0 #800000 2 1>
  <Line 0 -15 0 -5 #800000 2 1>
  <Line -10 -5 10 -10 #800000 2 1>
  <Line 0 -20 0 -10 #000080 2 1>
  <Line -10 5 10 10 #800000 2 1>
  <Line -6 15 6 0 #800000 2 1>
  <Line 0 9 0 11 #800000 2 1>
  <Line 0 20 0 10 #000080 2 1>
  <Line -10 0 20 0 #800000 2 1>
  <Line 10 0 10 0 #000080 2 1>
  <Line -10 -15 0 30 #800000 3 1>
  <Text 5 -10 5 #005500 0 "bn">
  <.PortSym 0 -30 1 90>
  <.PortSym -30 0 2 0>
  <.PortSym 0 30 3 270>
  <.PortSym 20 0 4 180>
  <.ID 10 10 X>
</Symbol>
```

- Directory structure of user lib:

```
~/.qucs/
└── user_lib
    ├── SG13G2_HBT
    │   └── SG13G2_HBT.sym
    └── SG13G2_HBT.lib*
```

* HSPICE models file from OpenPDK without any modifications

Resistor Models

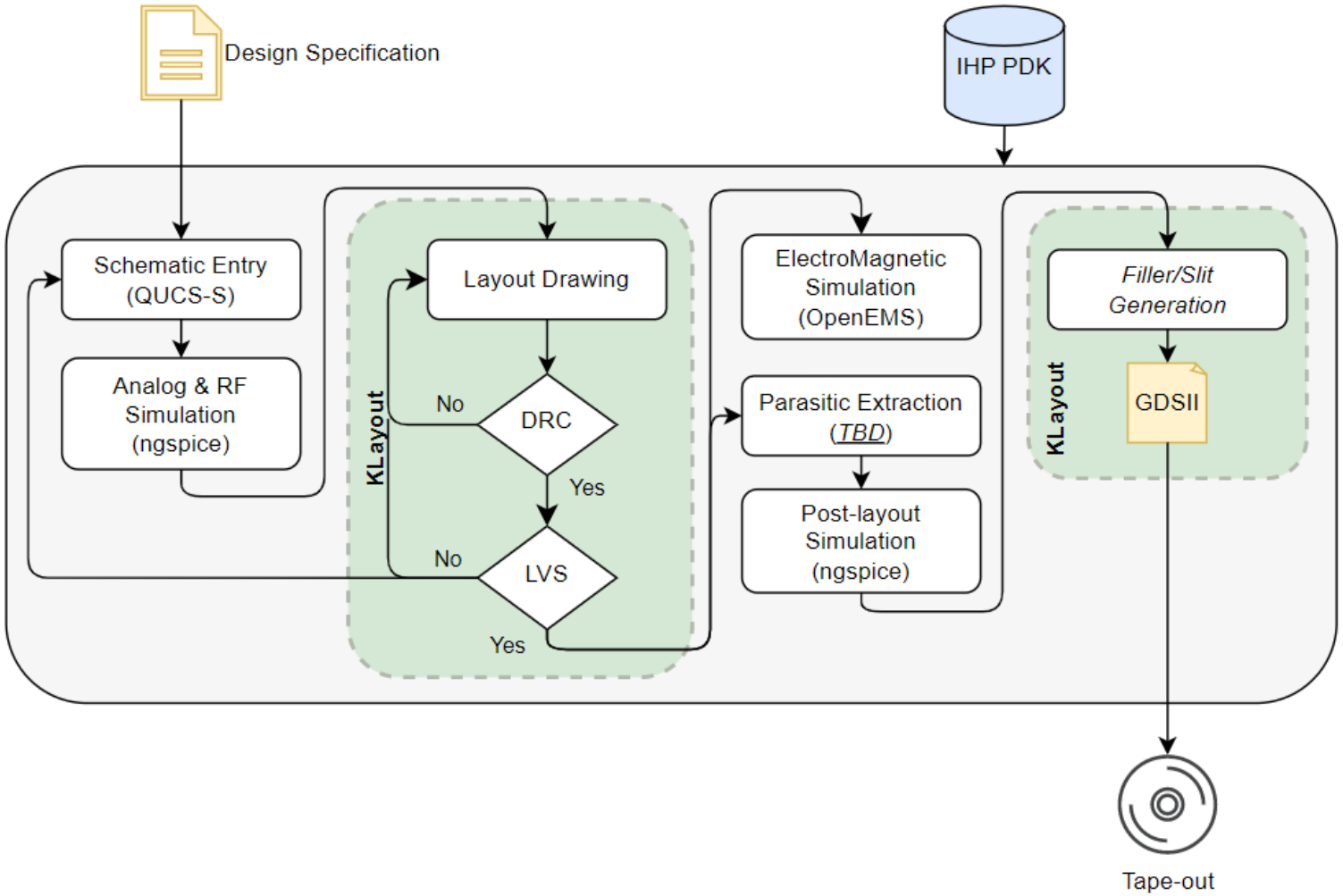
- Spice Models Ready for
  - Rsil (Rs= 7 Ω/▢)
  - Rhigh (Rs= 1360 Ω/▢)
  - Rppd (Rs= 260 Ω/▢)
- Temperature Modeling ✓
- Noise Modeling ✓
- Ngspice, Xyce compatible ✓
- Non-linear effects not included: ✗
  - Self heating
  - Velocity saturation
  - …
- Working on adapting **R3 CMC** resistor model.

Simple linear spice semiconductor resistor model

OpenPDK and OpenTooling – Status

1. IHP Open Source EDA Flows Proposals
2. OpenPDK Project on GitHub
3. Dedicated OpenPDK Virtual Linux Host Machine
4. Available OpenPDK Data
5. Open Questions
6. Next Steps / Planned Updates
7. Summary / Goals

Analog/RF OpenPDK/EDA Flow Proposal



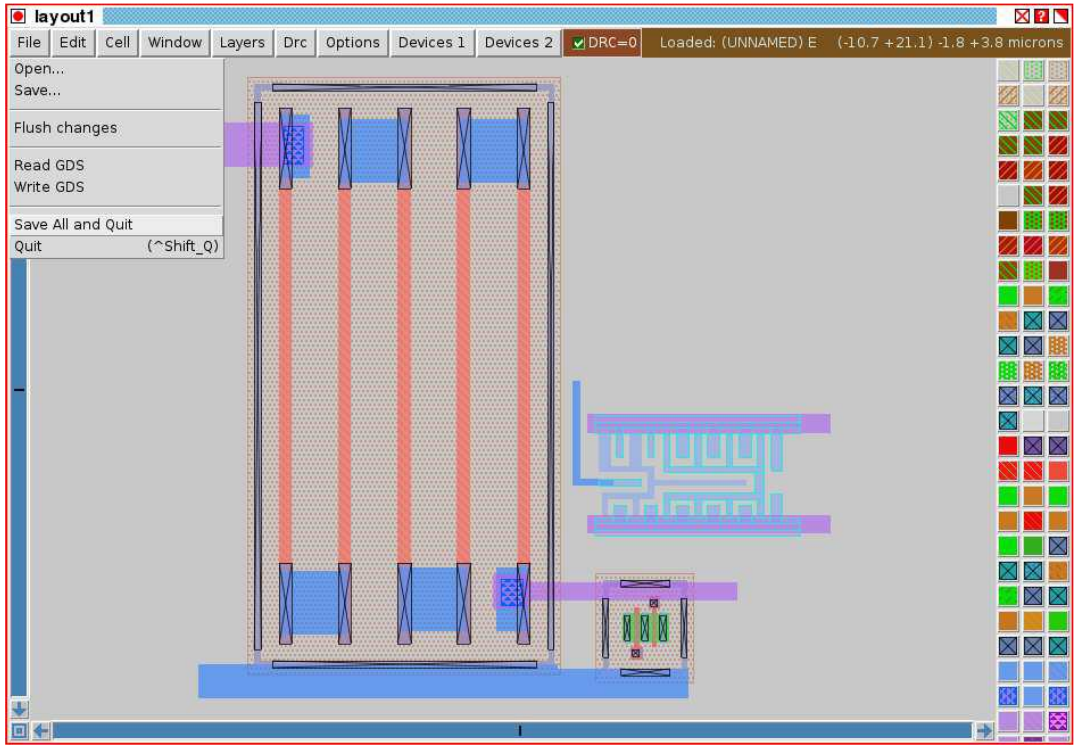
- KLayout-oriented flow
  - Layout design
  - Parameterizable cells
  - Physical Verification
- QUCS-S
- ngspice
- OpenEMS
- … ?

Layout Design: should we support Magic VLSI?

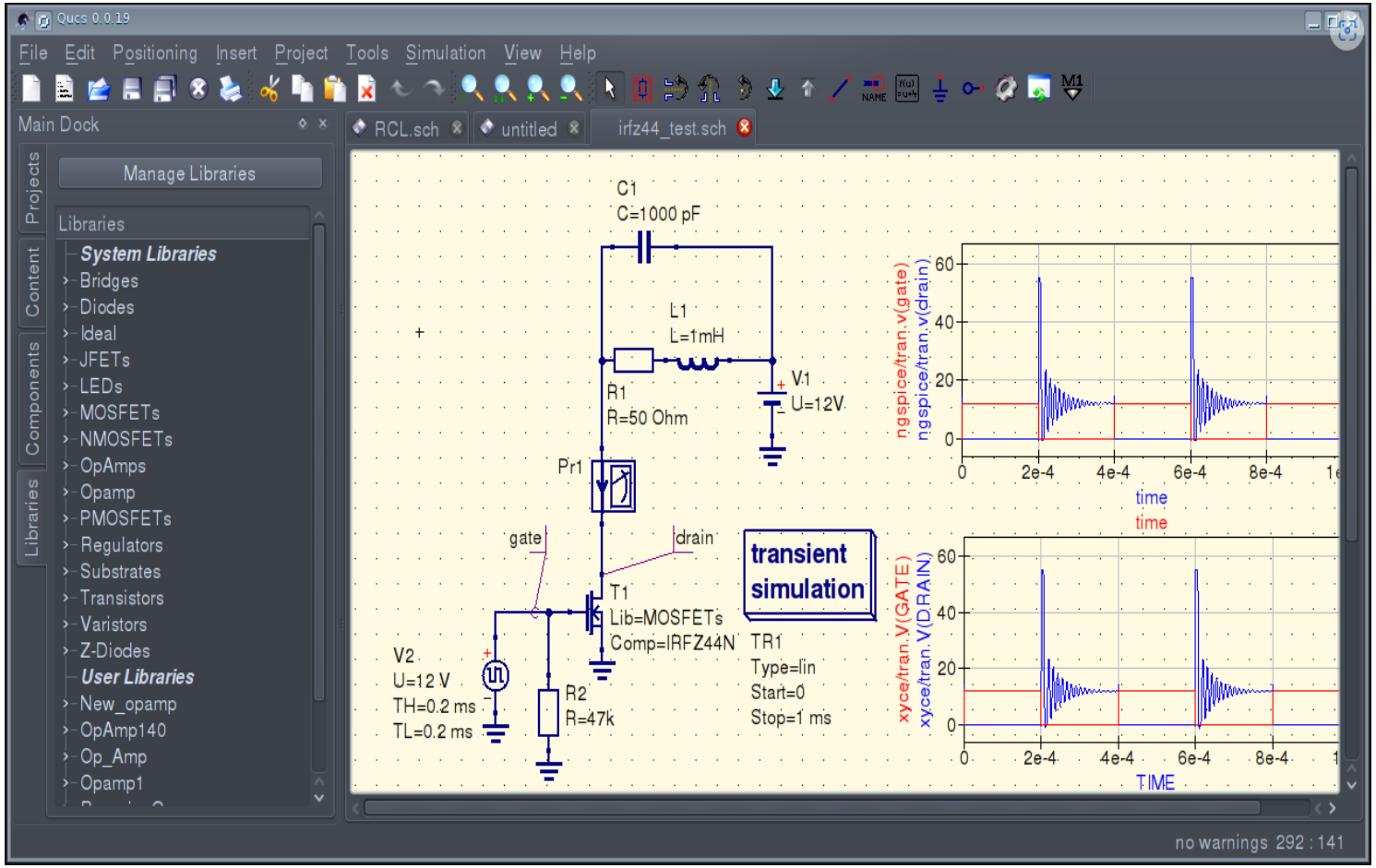
KLayout

```
# The PCell declaration for the circle
class StarPCell < PCellDeclarationHelper

  include RBA

  def initialize

    # Important: initialize the super class
    super

    # declare the parameters
    param(:l, TypeLayer, "Layer", :default => LayerDefs::new(1, 0))
    param(:r1, TypeDouble, "Inner radius", :default => 1, :unit => "µm")
    param(:r2, TypeDouble, "Outer radius", :default => 2, :unit => "µm")
    param(:n, TypeInt, "Number of rays", :default => 11)
    param(:da, TypeInt, "Ray angle", :default => 3, :unit => "deg")

  end

  def display_text_impl
    # Provide a descriptive text for the cell
    "StarPCell(L=#{l.to_s},R1=#{'%.3f' % r1.to_f},R2=#{'%.3f' % r2.to_f},N=#{n},Da=#{'%.3f' % da.to_f})"
  end

  def produce_impl

    # This is the main part of the implementation: create the layout

    # compute the ray parts and produce the polygon
    d = Math::PI * da / 180.0
    a = 0.0
    n.times do |i|
      dpts = [
        DPoint::new(r1 * Math.cos(a - d), r1 * Math.sin(a - d)),
        DPoint::new(r1 * Math.cos(a + d), r1 * Math.sin(a + d)),
        DPoint::new(r2 * Math.cos(a - d), r2 * Math.sin(a - d)),
        DPoint::new(r2 * Math.cos(a + d), r2 * Math.sin(a + d))
      ]
      cell.shapes(l_layer).insert(DPolygon::new(dpts))
      a += Math::PI * 2 / n
    end

  end

end
```

Magic

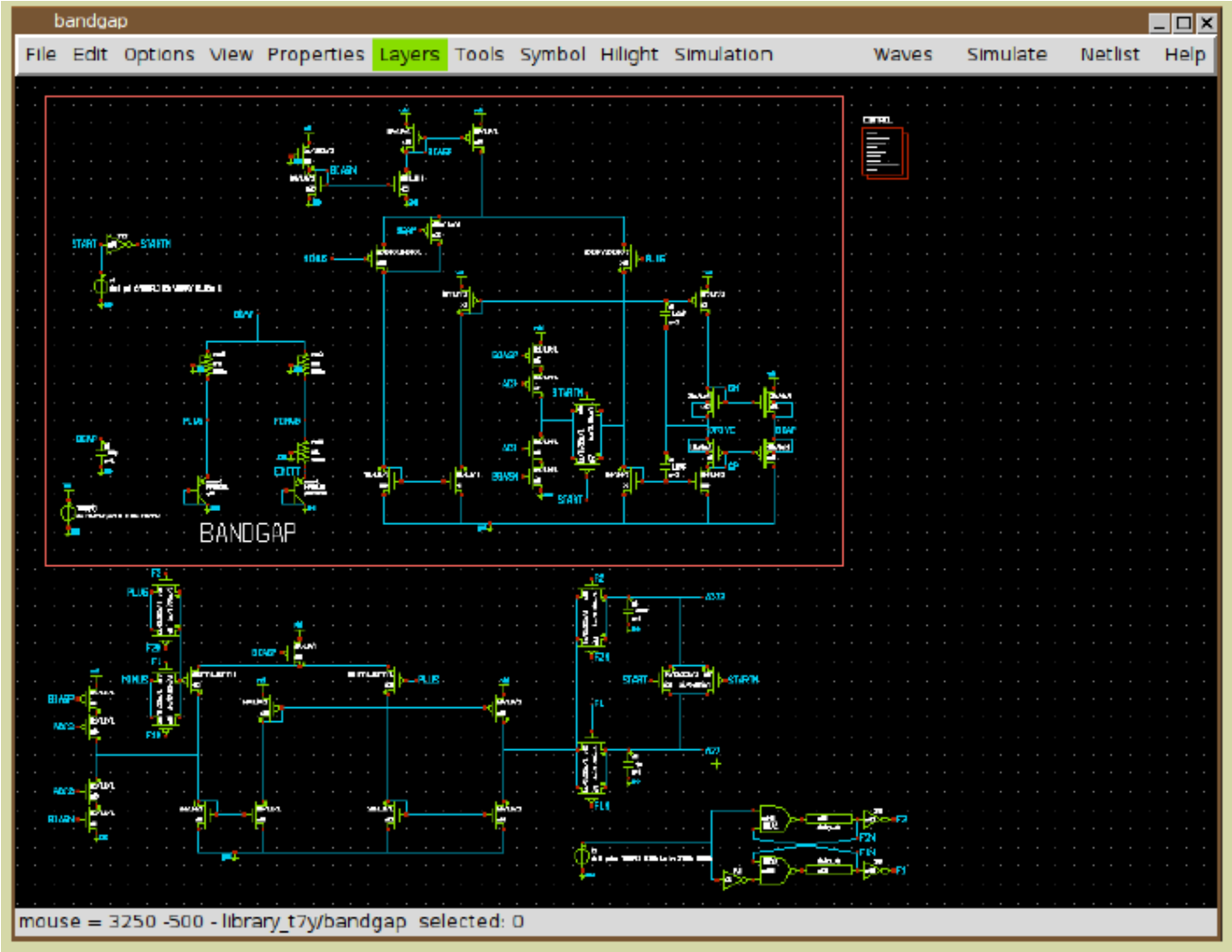


Circuit Design: QUCS-S / Xschem / Revolution EDA



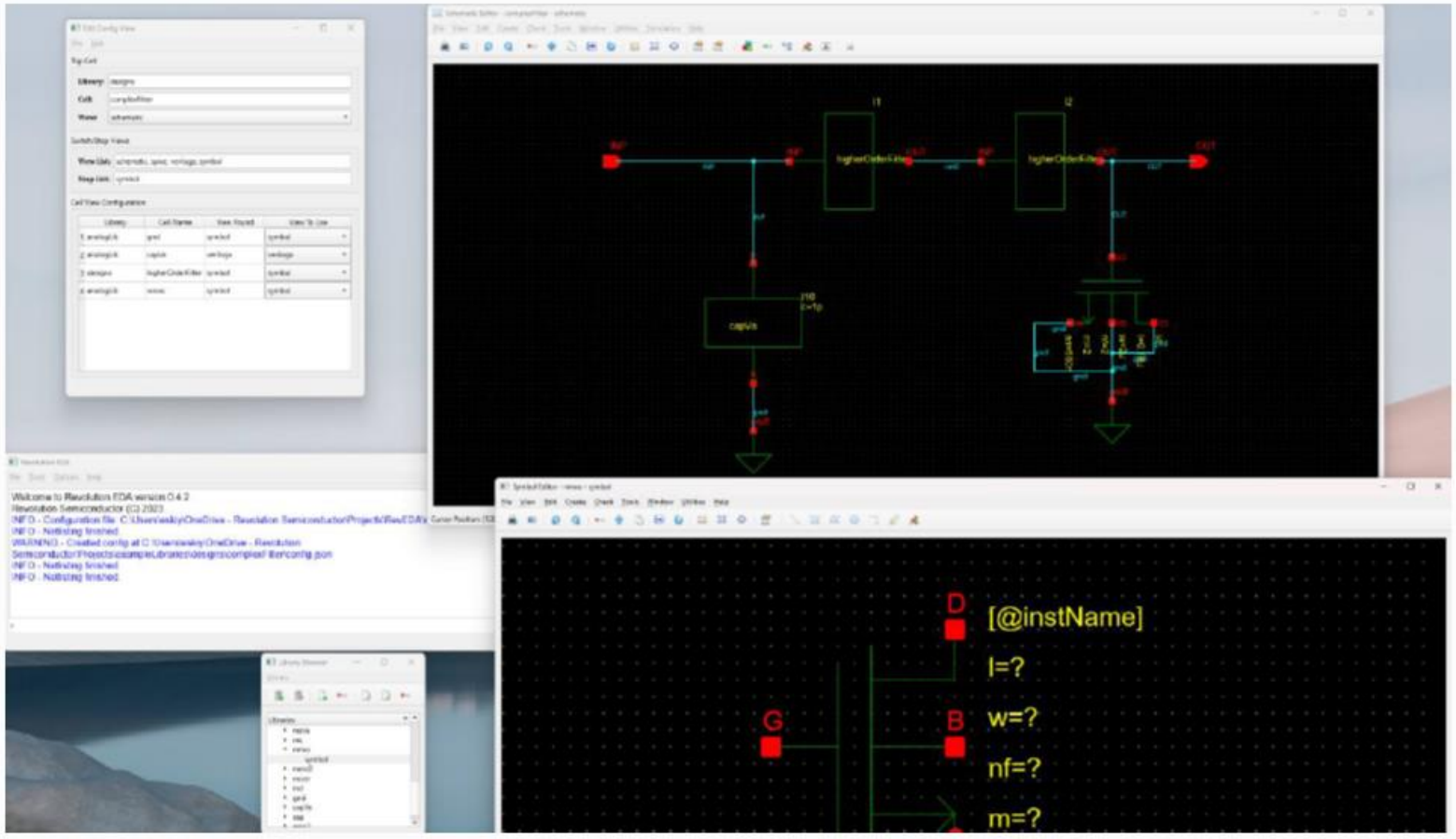
- Direct support of SPICE models from components datasheets
- Basic and Advanced SPICE components and simulations
- Direct support of SPICE Modelcards, SPICE sections (.IC, .NODESET)
- Parametric circuits (.PARAM) and SPICE postprocessor (Nutmeg)
- Single-tone and Multitone Harmonic balance analysis w/ XYCE backend
- Nutmeg script simulation
- XYCE script simulation type and digital device library
- XSPICE CodeModel synthesizer
- User mathematical functions definitions with .FUNC (added in 0.0.20)

Circuit Design: QUCS-S / Xschem(?) / Revolution EDA



- Hierarchical representation of circuits
- Generate circuit netlists for SPICE, Verilog, VHDL, tEDAx
- Components: primitives, behavioral blocks, subcircuit blocks
- True mixed mode circuit description: Analog, Behavioral, Transistor-level, Gate-level
- Efficient handling of Very large designs, no scripting language for intensive computations
- GUI and scripting language with Tcl-Tk
- Vector instances and bus notations like DATA[15:0,31:16]
- TCL API for forward / backward annotation to / from 3rd party EDA software
- Used for SkyWater OpenPDK

Circuit Design: QUCS-S / Xschem / Revolution EDA(?)



- Revolution EDA has robust symbol and schematic entry tools
  - All-around Python based.
  - Built-in Python console.
  - State-of-the-art QT6 toolkit.
  - Shared-source (MPL modified with Commons Clause)
- Big ambitions:
  - Layout Editor
  - Simulation GUI cockpit
  - AI in circuit design, and much more…

Simulation: should we support Xyce?

Ngspice Home

- Home
- What is ngspice ?
- Features, Extras & Options
- F.A.Q.
- Tutorials
- Introductory and update videos

Sourceforge Developer Pages

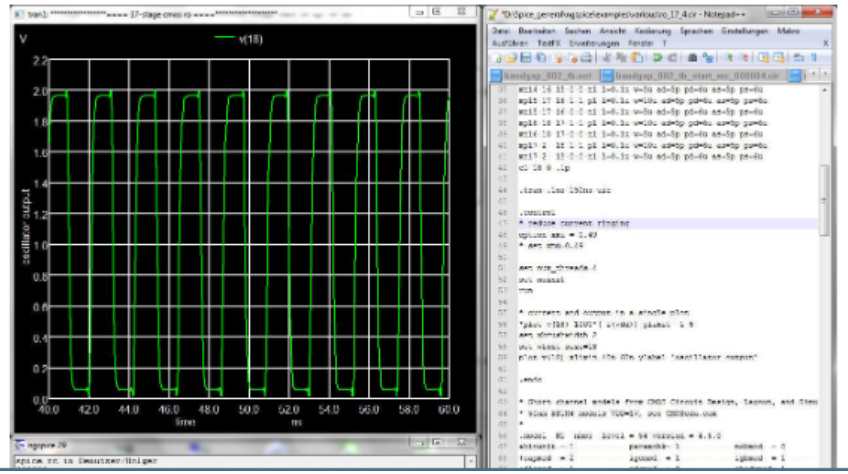
ngspice - open source spice simulator

ngspice is the open source spice simulator for electric and electronic circuits.

Such a circuit may comprise of JFETs, bipolar and MOS transistors, passive elements like R, L, or C, diodes, transmission lines and other devices, all interconnected in a netlist. Digital circuits are simulated as well, event driven and fast, from single gates to complex circuits. And you may enter the combination of both analog and digital as a mixed-signal circuit.

ngspice offers a wealth of device models for active, passive, analog, and digital elements. Model parameters are provided by our collections, by the semiconductor device manufacturers, or from semiconductor foundries. The user adds her circuits as a netlist, and the output is one or more graphs of currents, voltages and other electrical quantities or is saved in a data file.

ngspice does not provide schematic entry. Its input is command line or file based. There are however third party interfaces available.



OpenPDK and OpenTooling – Status

1. IHP Open Source EDA Flows Proposals
2. OpenPDK Project on GitHub
3. Dedicated OpenPDK Virtual Linux Host Machine
4. Available OpenPDK Data
5. Open Questions
6. Next Steps / Planned Updates
7. Summary / Goals

Next Steps / Planned Updates

- Start set of Digital cells
  - Additional cells in development by ETH Zurich + community
- LEF view for primitive devices
- KLayout Technology file
  - Add connectivity section
- **Need help on**:
  - **MOS SPICE models**
  - **PCells**

- Tasks on GitHub:

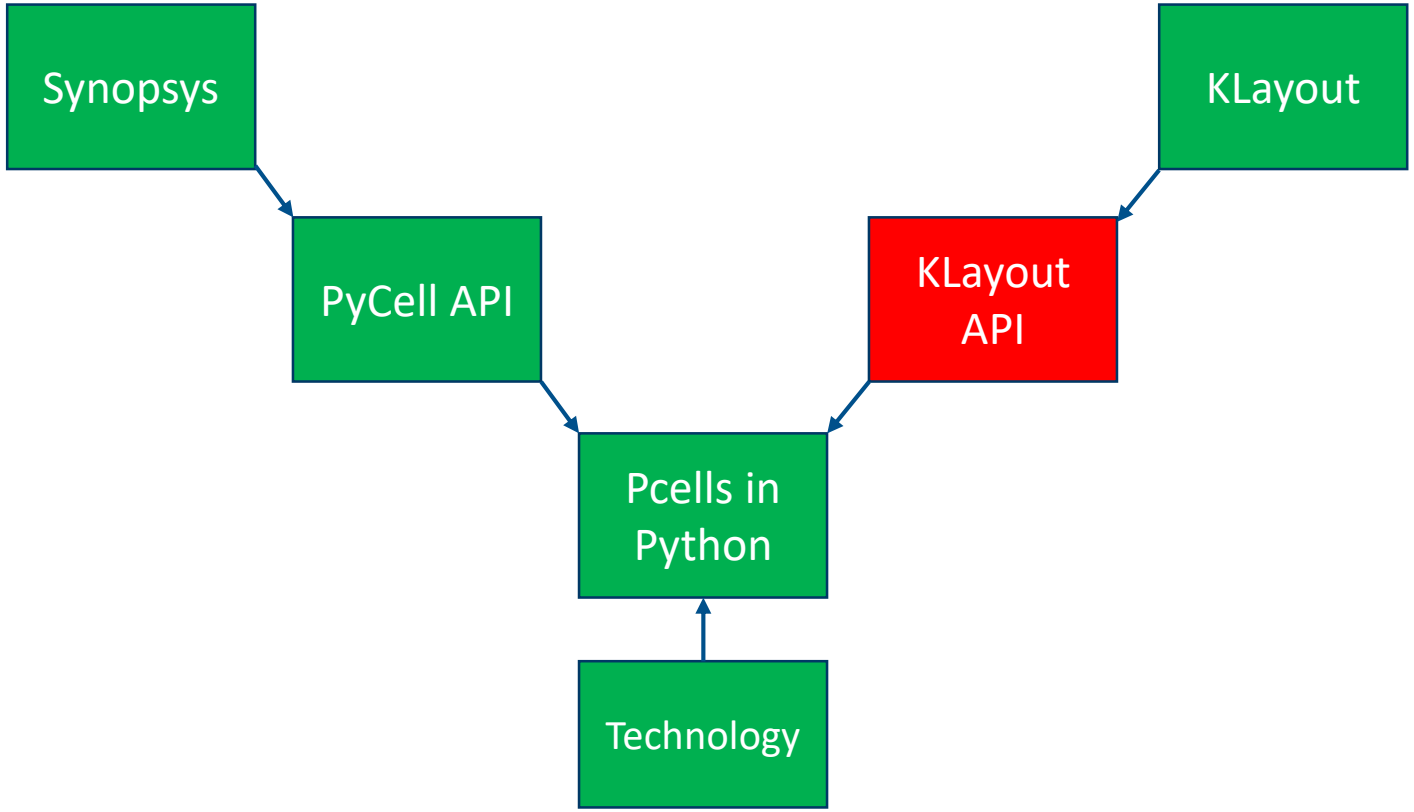
	Title	Status
1	Ngspice: 'ignored parameter' messages for HBT models	Done
2	Start set of Digital cells	Done
3	Digital standard cells enhancements (increased set)	In Progress
4	LEF view for primitive devices	Todo
5	MOS HSPICE models	Todo
6	KLayout Tech file	Done
7	DRM for Opensource PDK	Done
8	QUCS-S Library w/ IHP OpenPDK devices	In Progress
9	Move documentation to ReadTheDocs framework	Todo
10	PyCells	Todo

MOS SPICE Models

- SG13G2 MOS Spectre models to SPICE format conversion to use with ngspice/Xyce simulators
  - supported:
    - extreme value behavior, corner cases
    - transistor layout data transfer
    - drain/source area calculations using the fingers number
  - some Spectre commands/instructions/statements not supported:
    - geometry checks
    - static and dynamic states checks
    - statistical variations

OpenPDK Pcells

- SG13G2 Synopsys PyCells to KLayout Python Pcells conversion

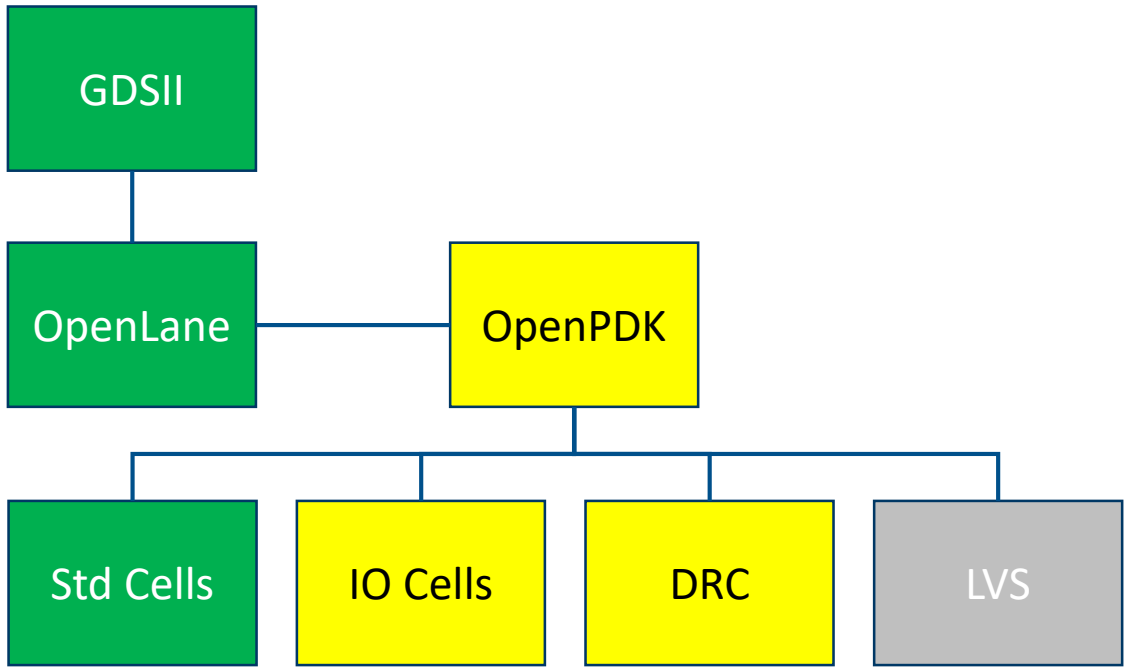


OpenPDK and OpenTooling – Status

1. IHP Open Source EDA Flows Proposals
2. OpenPDK Project on GitHub
3. Dedicated OpenPDK Virtual Linux Host Machine
4. Available OpenPDK Data
5. Open Questions
6. Next Steps / Planned Updates
7. Summary / Goals

Summary / Goals

- First phase goal ➔ Digital design submitted w/ core expert group in Dec Y23



Legend:
- available
- in progress
- optional

Acknowledgment

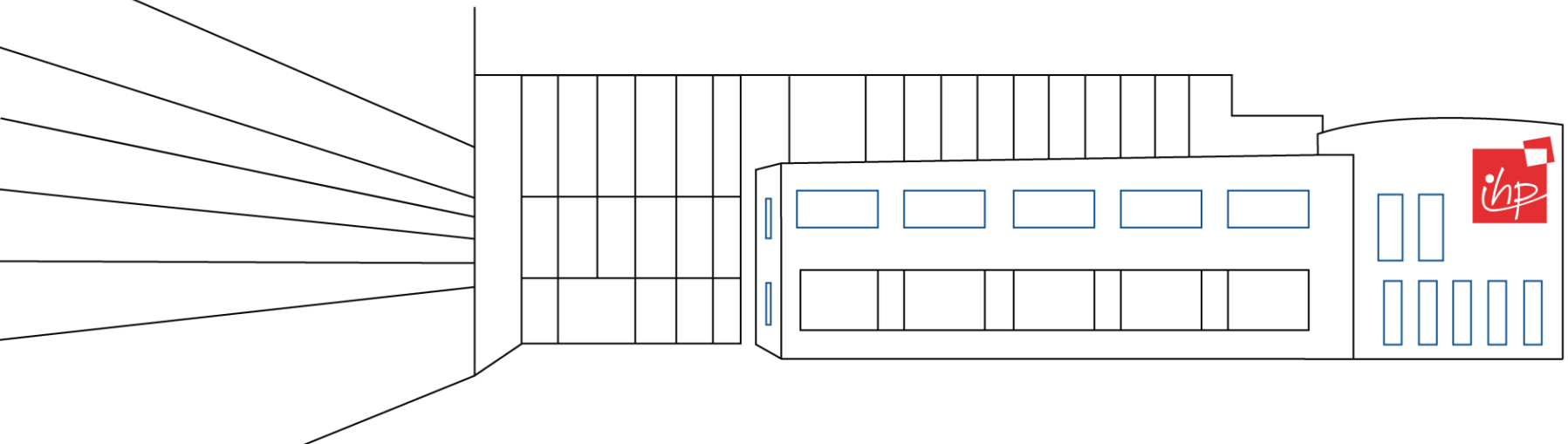
- Thanks to my colleagues at IHP
- Thanks to ETH Zurich + open source community
- Separate thanks to Volker Mühlhaus for work on the EM solvers
- And final thanks to different public founded German projects:
  - VE-HEP (16KIS1339K) https://elektronikforschung.de/projekte/ve-hep-1
  - IHP Open130-G2 (16ME0852) https://www.elektronikforschung.de/projekte/ihp-open130-g2
  - FMD-QNC (16ME0831) https://www.elektronikforschung.de/projekte/fmd-qnc
  - Workshop funding - FMD-QNC with VDI/VDE (project management agency) approval

SPONSORED BY THE
Federal Ministry of Education and Research

Forschungsfabrik Mikroelektronik Deutschland

Thank you for your attention!

IHP – Leibniz Institute for High Performance Microelectronics
Im Technologiepark 25
15236 Frankfurt (Oder)
Tel.: +49 (0) 335 5625 733
E-Mail: andreev@ihp-microelectronics.com



Forschungsfabrik Mikroelektronik Deutschland

Leibniz Association