オープン・ソース・EDA

FSIC 2025紹介

オープン・ソース・シリコン・タイムライン

2018：DARPA（国防高等研究計画局）OpenIDEA プログラム

2019：efabless/Google が SkyWater の PDK をオープン化

2020：Google/efabless/SkyWater OpenMPW プログラムスタート

2022：Global Foundries が OpenMPW プログラムに参加

2023：独) iHP (130nm/SiGe）が PDK のオープン化を宣言

2024：Google OpenMPW プログラムを停止

2025：efabless 会社清算

https://www.darpa.mil/program/intelligent-design-of-electronic-assets
https://www.darpa.mil/attachments/eri_design_proposers_day.pdf
https://github.com/The-OpenROAD-Project
https://developers.google.com/silicon

オープン・ソース・シリコン・イベントカレンダー

2012 : **ORCONF** (Open RISC-V conference) started

2015 : **FOSSi foundation** kicked off in ORCONF 2015 in Valencia, Spain

2019 : **FSiC 2019** kicked off on Sorbonne Université, Paris

2019 : Linux Foundation to host **Chips Alliance** in USA

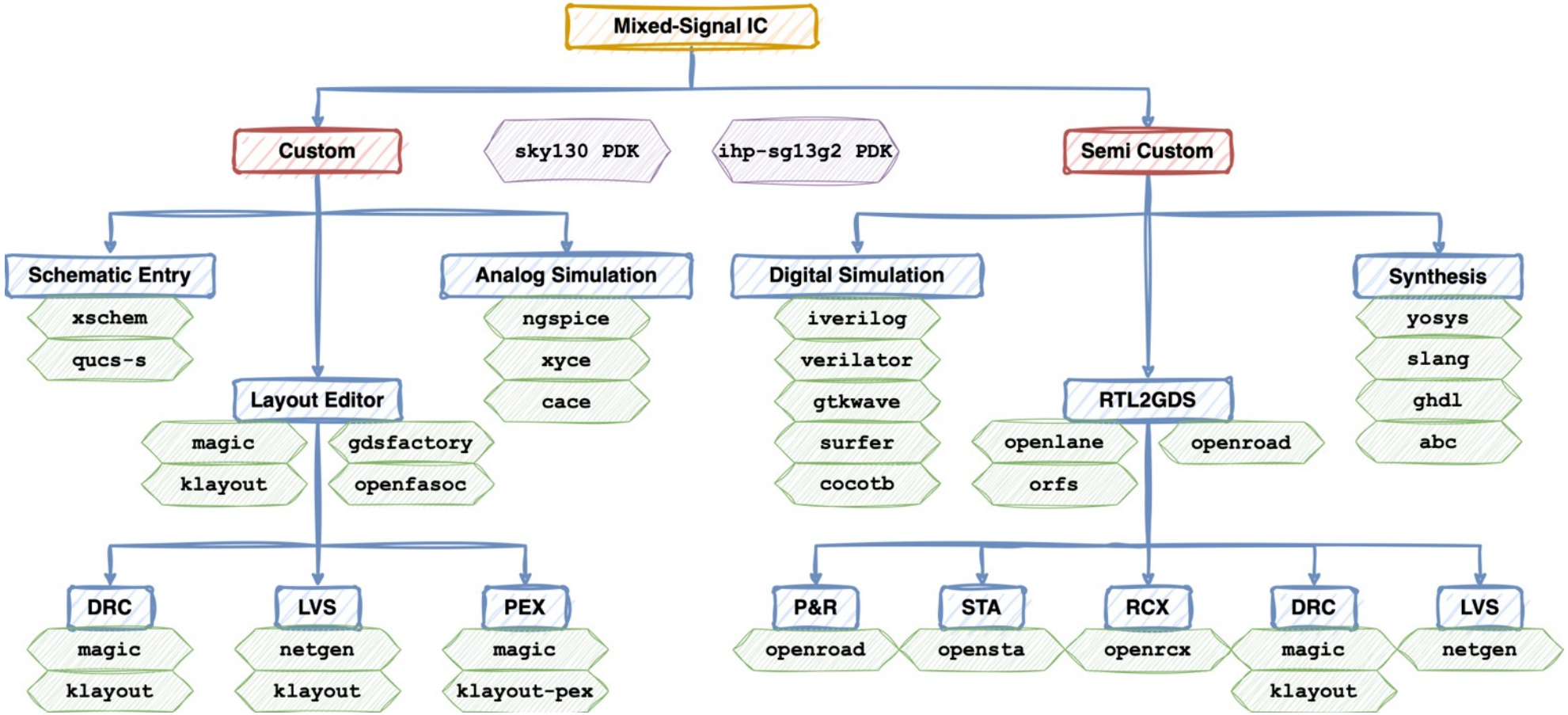
2019 : **Latch-up,** annual North America event hosted by FOSSi foundation

2024 : AIST established a non-profit **OpenSUSI** in Japan

2025 : China revealed **“一生一芯”** program in FSiC 2025

Lower the Barrier of Entry into Open-Source EDA

- **Docker**-based **integrated solution** available at https://github.com/iic-jku/IIC-OSIC-TOOLS



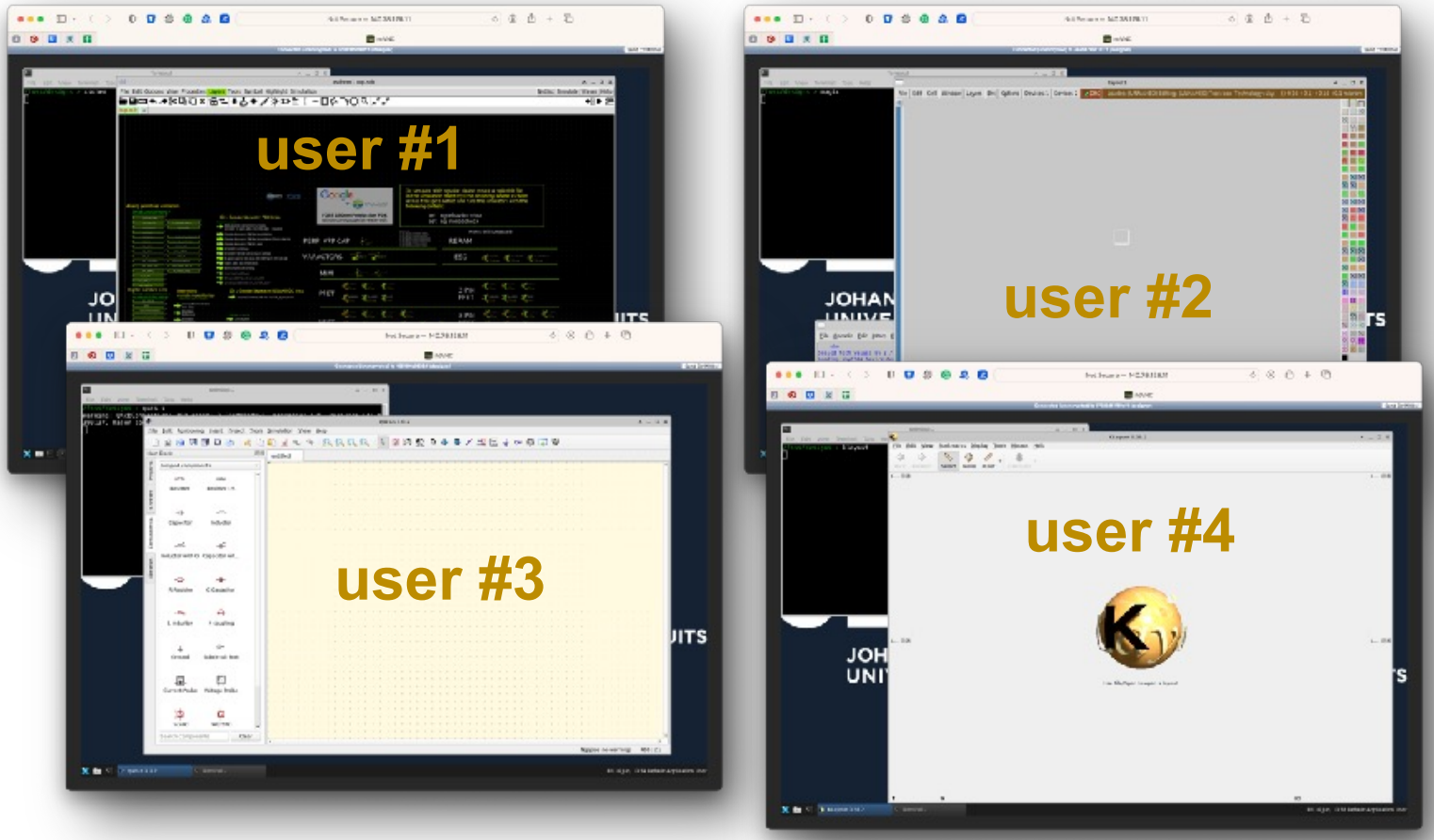
Only for ISHI会

How to Provide EDA Environments for Students?

- Tried different approaches (local installation on students' devices, providing virtual machines, etc.)
  ➔ time-consuming
- **Current solution:**
  - Provide **VMs** on university servers, access via **browser**
  - Option: **local install** (VM using Docker) for advanced students (Win, Linux, macOS)

➔ All students on exactly same setup

- **IIC-OSIC-TOOLS** readily deployed in cloud/servers (using Docker and simple scripts)




```
./eda_server_start.sh -p 50010 -n 4 -m harald-demo -f harald_demo.json
```


Only for ISHI会

IIC-OSIC-TOOLS Fact Sheet

- **Monthly releases** since May'2022 (latest is 2025.05)
  - No rolling releases (yet?) ➔ too much tool interdependencies
- Runs on **Windows**, **Linux**, **macOS** // **x86_64/amd64** // **aarch64/arm64**
  - RISC-V planned for the future
- **Virtual machine (VM)** based on **Docker**
  - Abstract and isolate user's OS and environment (otherwise too much variability)
- Usage models: **VNC, browser, X11/Wayland, shell/headless, Jupyter server**
- Support via GitHub issues, pull requests (➔ community to help define features)
- DOI https://doi.org/10.5281/zenodo.14387234

End-to-end Open-Source IC Design is possible today!

Design: from PULP

github.com/pulp-platform

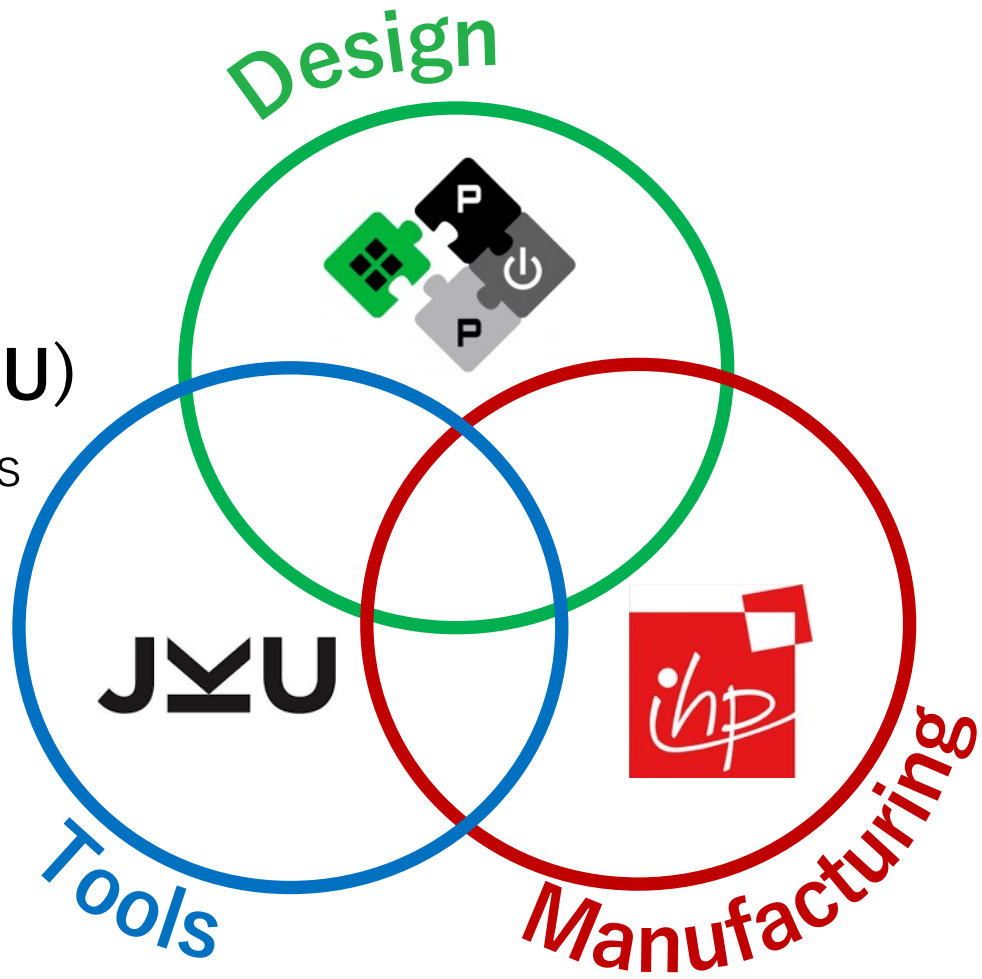
Tools: from Johannes Kepler University (JKU)

Reliable VM with large collection of open-source tools

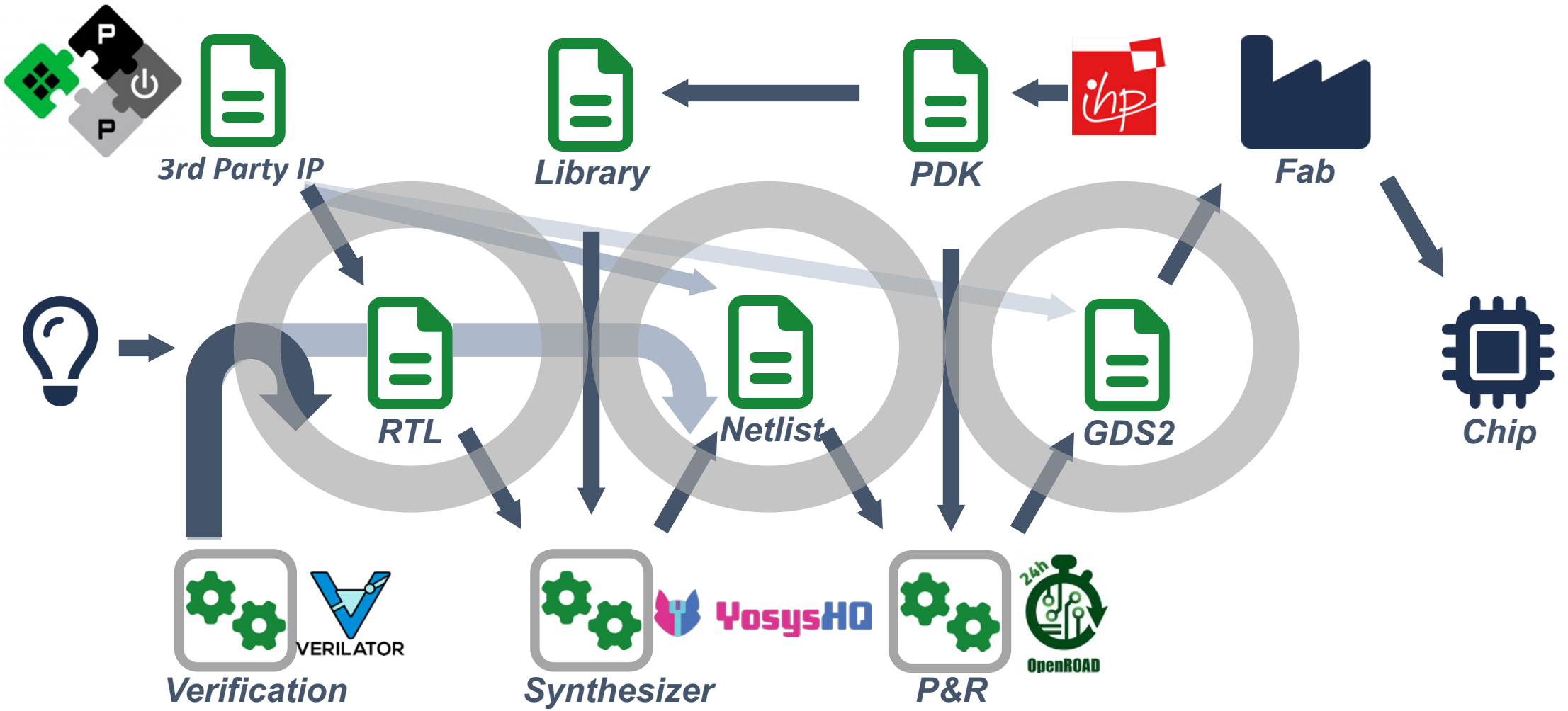
github.com/iic-jku/IIC-OSIC-TOOLS

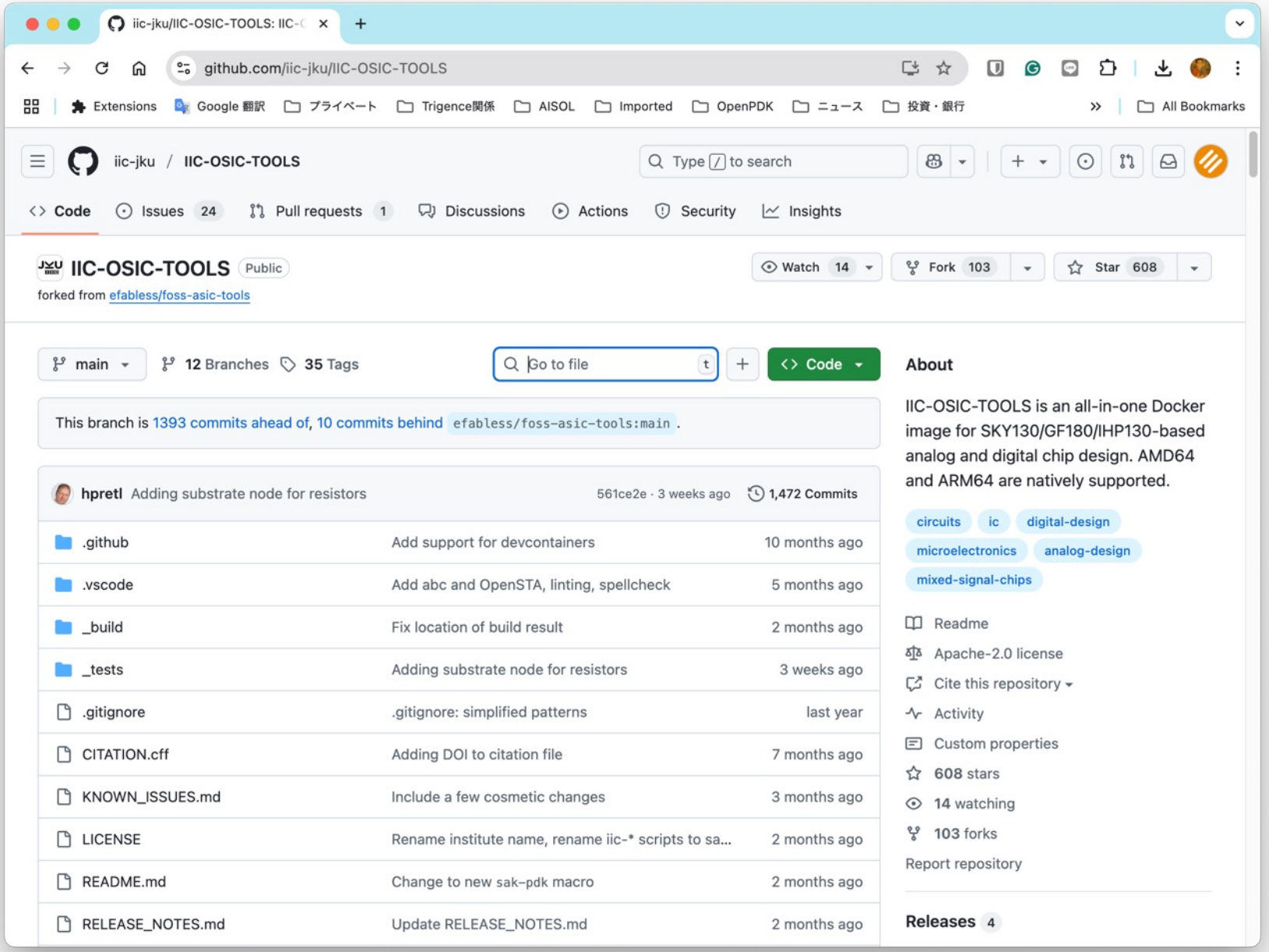
Manufacturing: IHP130nm

github.com/IHP-GmbH/IHP-Open-PDK



End-to-end Open-Source allows sharing of design data



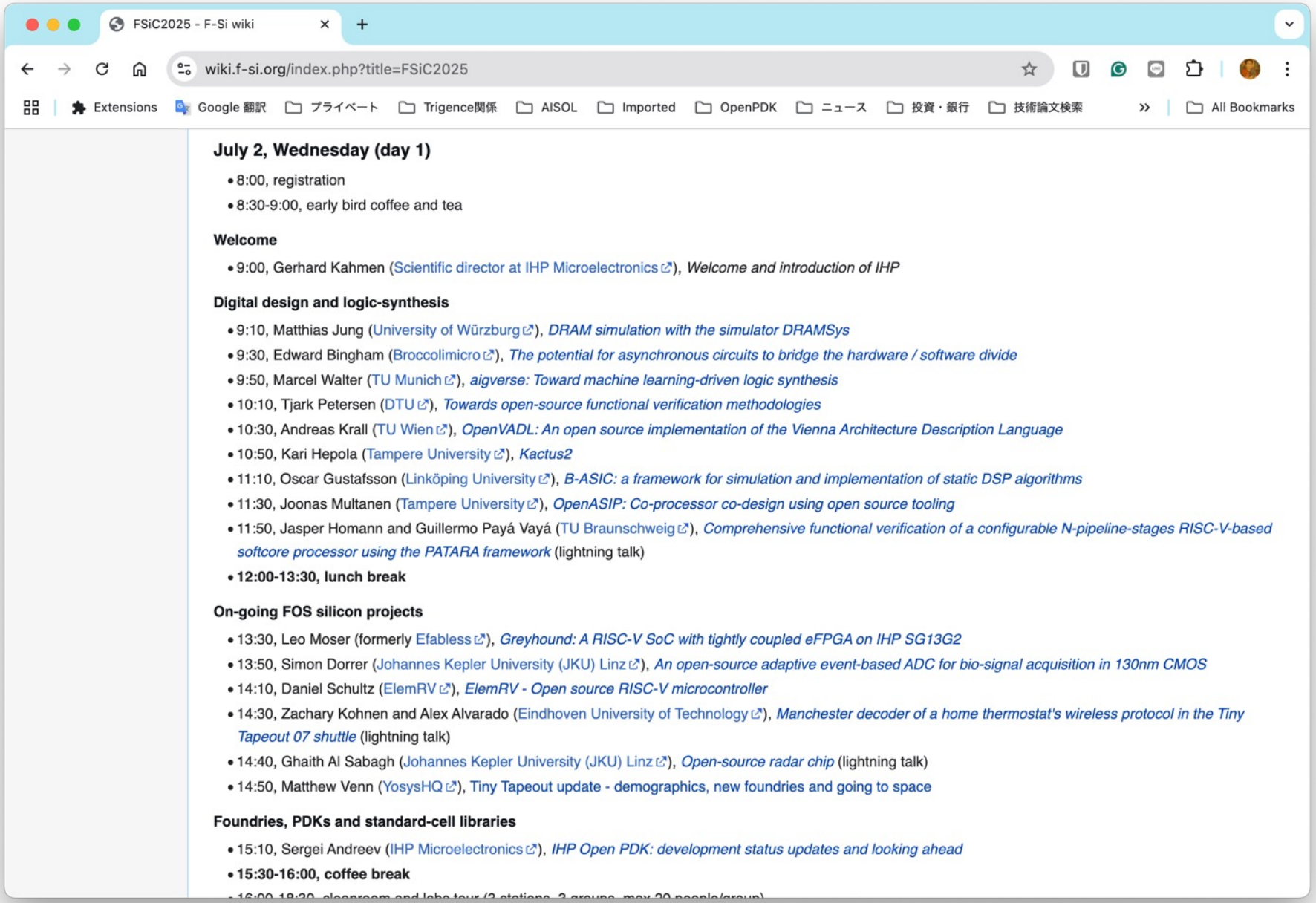


3. Installed Tools

Below is a list of the current too

abc sequential logic synthesis and forr
amaranth a Python-based HDL tool ch
cace a Python-based circuit automatic
cocotb simulation library for writing VH
covered Verilog code coverage
cvc circuit validity checker (ERC)
edalize Python abstraction library for E
fusesoc package manager and build to
gaw3-xschem waveform plot tool for x
gds3d a 3D viewer for GDS files
gdsfactory Python library for GDS gen
gdspy Python module for the creation
gf180mcu GlobalFoundries 180 nm CM
ghdl-yosys-plugin VHDL-plugin for yos
ghdl VHDL simulator
gtkwave waveform plot tool for digital s
hdl21 analog hardware description libr
ihp-sg13g2 IHP Microelectronics 130 r
yet; xschem and ngspice simulation w
irsim switch-level digital simulator
iverilog Verilog simulator
klayout-pex parasitic extraction for kla
klayout layout viewer and editor for GD
lctime Characterization kit for CMOS c
libman design library manager to mana
magic layout editor with DRC and PEX
netgen netlist comparison (LVS)
ngspice SPICE analog and mixed-sign
ngspyce Python bindings for ngspice
nvc VHDL simulator and compiler
open_pdks PDK setup scripts
openlane2 rewrite of OpenLane in Pytl
openram OpenRAM Python library
openroad RTL2GDS engine used by o
opensta gate level static timing verifier
openvaf Verilog-A compiler for device
osic-multitool collection of useful script
padring padring generation tool
pulp-tools PULP platform tools consist
pygmid Python version of the gm/Id sta
pyopus simulation runner and optimiza
pyrtl collection of classes for pythonic
pyspice interface ngspice and xyce fro
pyuvm Universal Verification Methodol
using cocotb
pyverilog Python toolkit for Verilog
qflow collection of useful conversion to
qucs-s simulation environment with RF
rggen code generation tool for configuration and
risc-v toolchain GNU compiler toolchain for RIS
riscv-pk RISC-V proxy kernel and bootloader
schemdraw Python package for drawing electric
siliconcompiler modular build system for hardwa
sky130 SkyWater Technologies 130 nm CMOS
slang yosys plugin Slang-based plugin for yosys
slang SystemVerilog parsing and translation (e.
spicelib library to interact with SPICE-like simula
spyci analyze/plot ngspice/xyce output data with
surelog SystemVerilog parser, elaborator, and U
surfer waveform viewer with snappy usable inte
verilator fast Verilog simulator
vlog2verilog Verilog file conversion
vlsirtools interchange formats for chip design.
volare version manager (and builder) for open-s
xcircuit schematic editor
xschem schematic editor
xyce fast parallel SPICE simulator (incl. xdm ne
yosys Verilog synthesis tool (with GHDL plugin
SystemVerilog synthesis), incl. eqy (equivalence
and mcy (mutation coverage)
RF toolkit with FastHenry2, FasterCap, openEM

FSiC 2025/Day 1st



July 2, Wednesday (day 1)

- 8:00, registration
- 8:30-9:00, early bird coffee and tea

Welcome

- 9:00, Gerhard Kahmen (Scientific director at IHP Microelectronics), *Welcome and introduction of IHP*

Digital design and logic-synthesis

- 9:10, Matthias Jung (University of Würzburg), *DRAM simulation with the simulator DRAMSys*
- 9:30, Edward Bingham (Broccolimicro), *The potential for asynchronous circuits to bridge the hardware / software divide*
- 9:50, Marcel Walter (TU Munich), *aigverse: Toward machine learning-driven logic synthesis*
- 10:10, Tjark Petersen (DTU), *Towards open-source functional verification methodologies*
- 10:30, Andreas Krall (TU Wien), *OpenVADL: An open source implementation of the Vienna Architecture Description Language*
- 10:50, Kari Hepola (Tampere University), *Kactus2*
- 11:10, Oscar Gustafsson (Linköping University), *B-ASIC: a framework for simulation and implementation of static DSP algorithms*
- 11:30, Joonas Multanen (Tampere University), *OpenASIP: Co-processor co-design using open source tooling*
- 11:50, Jasper Homann and Guillermo Payá Vayá (TU Braunschweig), *Comprehensive functional verification of a configurable N-pipeline-stages RISC-V-based softcore processor using the PATARA framework* (lightning talk)
- **12:00-13:30, lunch break**

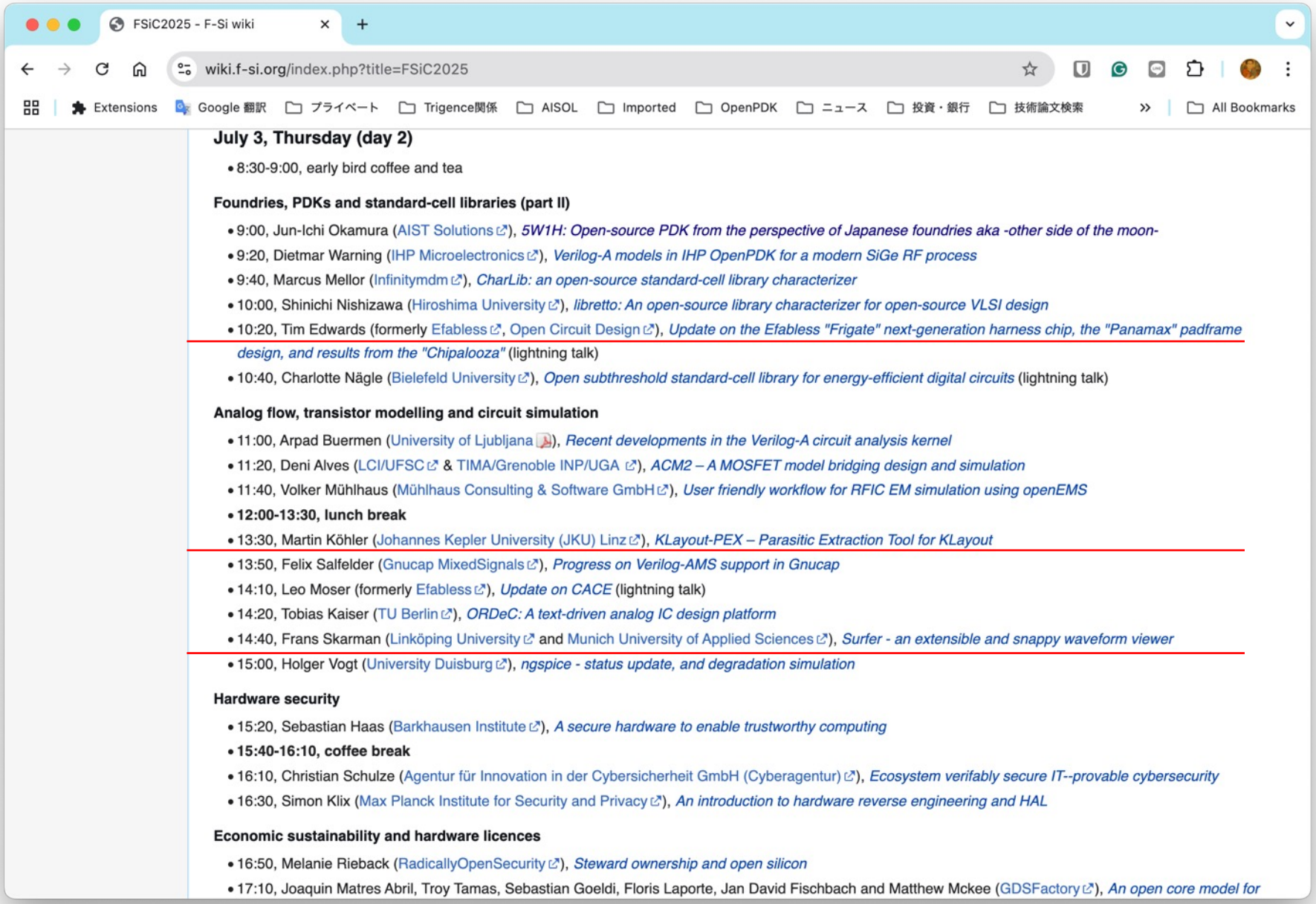
On-going FOS silicon projects

- 13:30, Leo Moser (formerly Efabless), *Greyhound: A RISC-V SoC with tightly coupled eFPGA on IHP SG13G2*
- 13:50, Simon Dorrer (Johannes Kepler University (JKU) Linz), *An open-source adaptive event-based ADC for bio-signal acquisition in 130nm CMOS*
- 14:10, Daniel Schultz (ElemRV), *ElemRV - Open source RISC-V microcontroller*
- 14:30, Zachary Kohnen and Alex Alvarado (Eindhoven University of Technology), *Manchester decoder of a home thermostat's wireless protocol in the Tiny Tapeout 07 shuttle* (lightning talk)
- 14:40, Ghaith Al Sabagh (Johannes Kepler University (JKU) Linz), *Open-source radar chip* (lightning talk)
- 14:50, Matthew Venn (YosysHQ), Tiny Tapeout update - demographics, new foundries and going to space

Foundries, PDKs and standard-cell libraries

- 15:10, Sergei Andreev (IHP Microelectronics), *IHP Open PDK: development status updates and looking ahead*
- **15:30-16:00, coffee break**

FSiC 2025/Day 2nd



July 3, Thursday (day 2)

- 8:30-9:00, early bird coffee and tea

Foundries, PDKs and standard-cell libraries (part II)

- 9:00, Jun-Ichi Okamura (AIST Solutions), *5W1H: Open-source PDK from the perspective of Japanese foundries aka -other side of the moon-*
- 9:20, Dietmar Warning (IHP Microelectronics), *Verilog-A models in IHP OpenPDK for a modern SiGe RF process*
- 9:40, Marcus Mellor (Infinitymdm), *CharLib: an open-source standard-cell library characterizer*
- 10:00, Shinichi Nishizawa (Hiroshima University), *libretto: An open-source library characterizer for open-source VLSI design*
- 10:20, Tim Edwards (formerly Efabless, Open Circuit Design), *Update on the Efabless "Frigate" next-generation harness chip, the "Panamax" padframe design, and results from the "Chipalooza"* (lightning talk)
- 10:40, Charlotte Nägle (Bielefeld University), *Open subthreshold standard-cell library for energy-efficient digital circuits* (lightning talk)

Analog flow, transistor modelling and circuit simulation

- 11:00, Arpad Buermen (University of Ljubljana), *Recent developments in the Verilog-A circuit analysis kernel*
- 11:20, Deni Alves (LCI/UFSC & TIMA/Grenoble INP/UGA), *ACM2 – A MOSFET model bridging design and simulation*
- 11:40, Volker Mühlhaus (Mühlhaus Consulting & Software GmbH), *User friendly workflow for RFIC EM simulation using openEMS*
- **12:00-13:30, lunch break**
- 13:30, Martin Köhler (Johannes Kepler University (JKU) Linz), *KLayout-PEX – Parasitic Extraction Tool for KLayout*
- 13:50, Felix Salfelder (Gnucap MixedSignals), *Progress on Verilog-AMS support in Gnucap*
- 14:10, Leo Moser (formerly Efabless), *Update on CACE* (lightning talk)
- 14:20, Tobias Kaiser (TU Berlin), *ORDeC: A text-driven analog IC design platform*
- 14:40, Frans Skarman (Linköping University and Munich University of Applied Sciences), *Surfer - an extensible and snappy waveform viewer*
- 15:00, Holger Vogt (University Duisburg), *ngspice - status update, and degradation simulation*

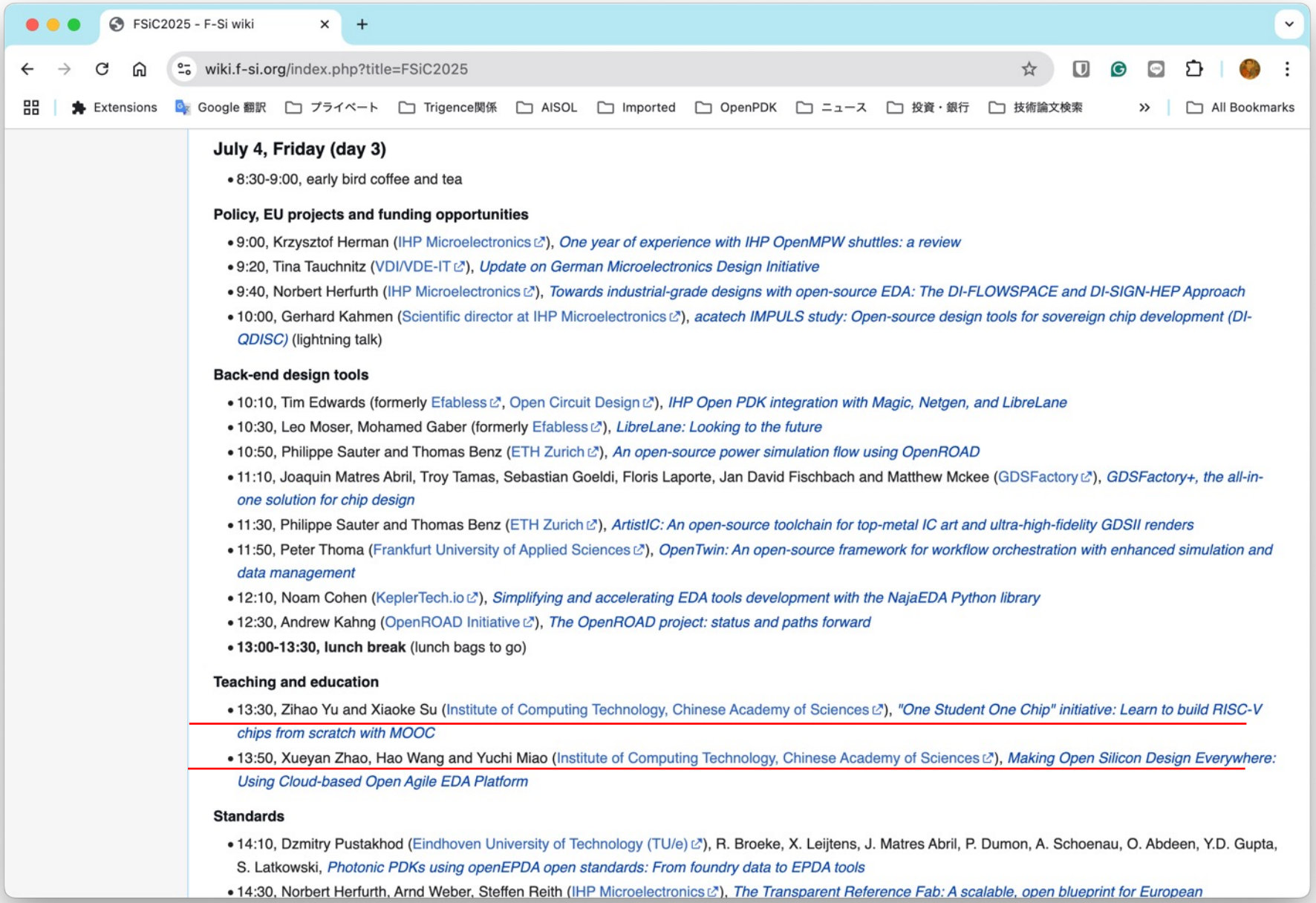
Hardware security

- 15:20, Sebastian Haas (Barkhausen Institute), *A secure hardware to enable trustworthy computing*
- **15:40-16:10, coffee break**
- 16:10, Christian Schulze (Agentur für Innovation in der Cybersicherheit GmbH (Cyberagentur)), *Ecosystem verifably secure IT--provable cybersecurity*
- 16:30, Simon Klix (Max Planck Institute for Security and Privacy), *An introduction to hardware reverse engineering and HAL*

Economic sustainability and hardware licences

- 16:50, Melanie Rieback (RadicallyOpenSecurity), *Steward ownership and open silicon*
- 17:10, Joaquin Matres Abril, Troy Tamas, Sebastian Goeldi, Floris Laporte, Jan David Fischbach and Matthew Mckee (GDSFactory), *An open core model for*

FSiC 2025/Day 3rd



July 4, Friday (day 3)

- 8:30-9:00, early bird coffee and tea

Policy, EU projects and funding opportunities

- 9:00, Krzysztof Herman (IHP Microelectronics), *One year of experience with IHP OpenMPW shuttles: a review*
- 9:20, Tina Tauchnitz (VDI/VDE-IT), *Update on German Microelectronics Design Initiative*
- 9:40, Norbert Herfurth (IHP Microelectronics), *Towards industrial-grade designs with open-source EDA: The DI-FLOWSPACE and DI-SIGN-HEP Approach*
- 10:00, Gerhard Kahmen (Scientific director at IHP Microelectronics), *acatech IMPULS study: Open-source design tools for sovereign chip development (DI-QDISC)* (lightning talk)

Back-end design tools

- 10:10, Tim Edwards (formerly Efabless, Open Circuit Design), *IHP Open PDK integration with Magic, Netgen, and LibreLane*
- 10:30, Leo Moser, Mohamed Gaber (formerly Efabless), *LibreLane: Looking to the future*
- 10:50, Philippe Sauter and Thomas Benz (ETH Zurich), *An open-source power simulation flow using OpenROAD*
- 11:10, Joaquin Matres Abril, Troy Tamas, Sebastian Goeldi, Floris Laporte, Jan David Fischbach and Matthew Mckee (GDSFactory), *GDSFactory+, the all-in-one solution for chip design*
- 11:30, Philippe Sauter and Thomas Benz (ETH Zurich), *ArtistIC: An open-source toolchain for top-metal IC art and ultra-high-fidelity GDSII renders*
- 11:50, Peter Thoma (Frankfurt University of Applied Sciences), *OpenTwin: An open-source framework for workflow orchestration with enhanced simulation and data management*
- 12:10, Noam Cohen (KeplerTech.io), *Simplifying and accelerating EDA tools development with the NajaEDA Python library*
- 12:30, Andrew Kahng (OpenROAD Initiative), *The OpenROAD project: status and paths forward*
- **13:00-13:30, lunch break** (lunch bags to go)

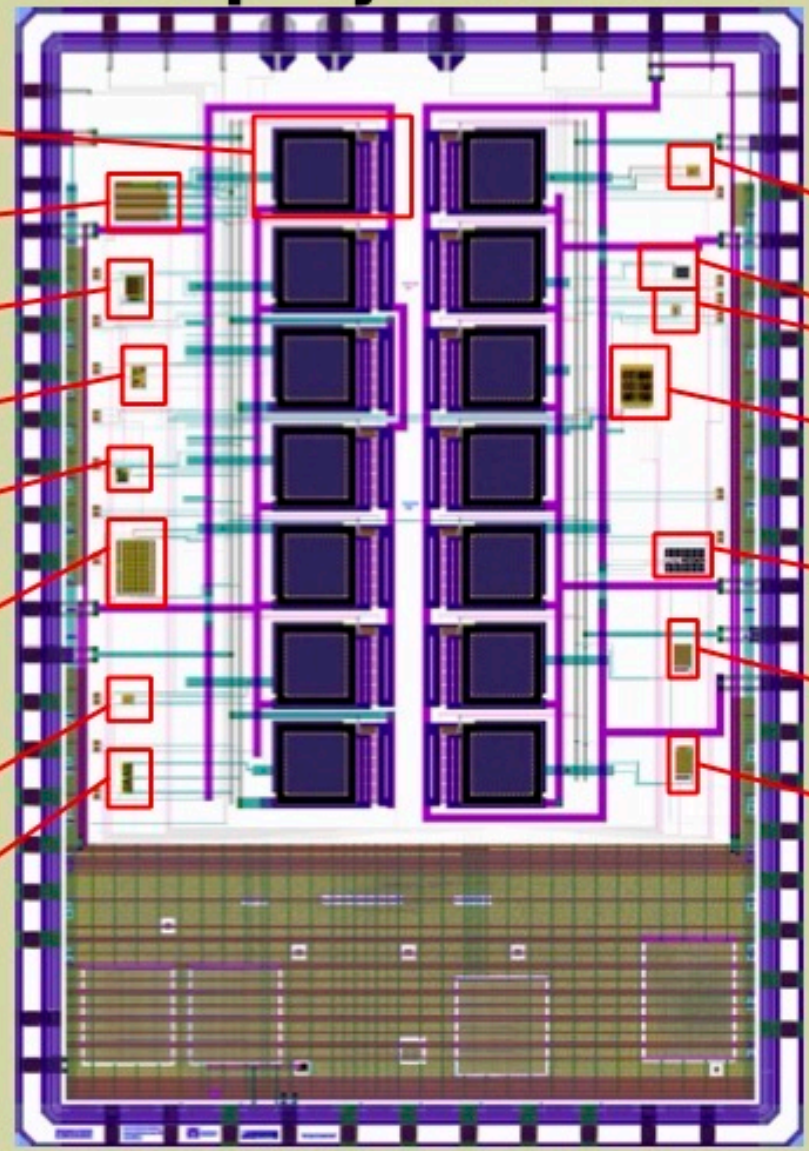
Teaching and education

- 13:30, Zihao Yu and Xiaoke Su (Institute of Computing Technology, Chinese Academy of Sciences), *"One Student One Chip" initiative: Learn to build RISC-V chips from scratch with MOOC*
- 13:50, Xueyan Zhao, Hao Wang and Yuchi Miao (Institute of Computing Technology, Chinese Academy of Sciences), *Making Open Silicon Design Everywhere: Using Cloud-based Open Agile EDA Platform*

Standards

- 14:10, Dzmitry Pustakhod (Eindhoven University of Technology (TU/e)), R. Broeke, X. Leijtens, J. Matres Abril, P. Dumon, A. Schoenau, O. Abdeen, Y.D. Gupta, S. Latkowski, *Photonic PDKs using openEPDA open standards: From foundry data to EPDA tools*
- 14:30, Norbert Herfurth, Arnd Weber, Steffen Reith (IHP Microelectronics), *The Transparent Reference Fab: A scalable, open blueprint for European*

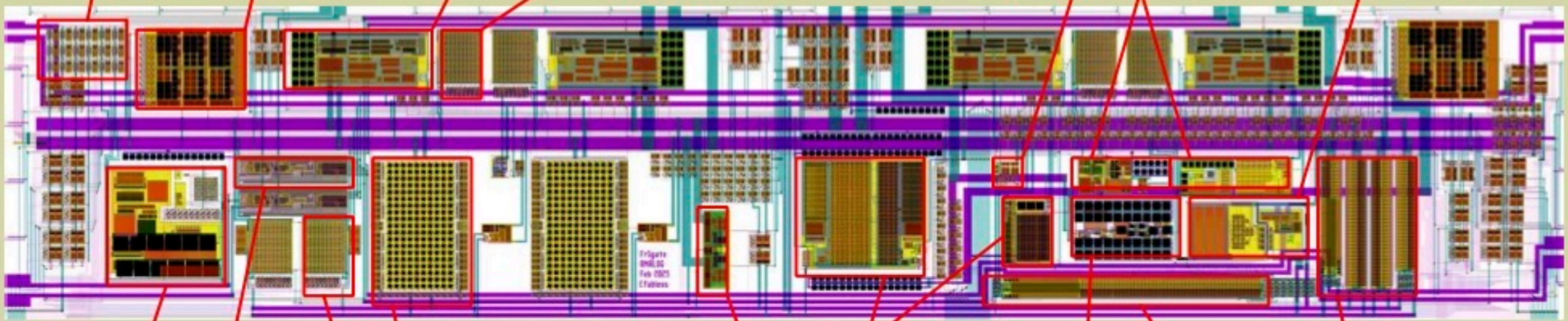
Chipalooza projects test chip #2



Original tapeout in November—stuck in fabrication.

“Frigate”: Open Source Mixed-Signal Harness

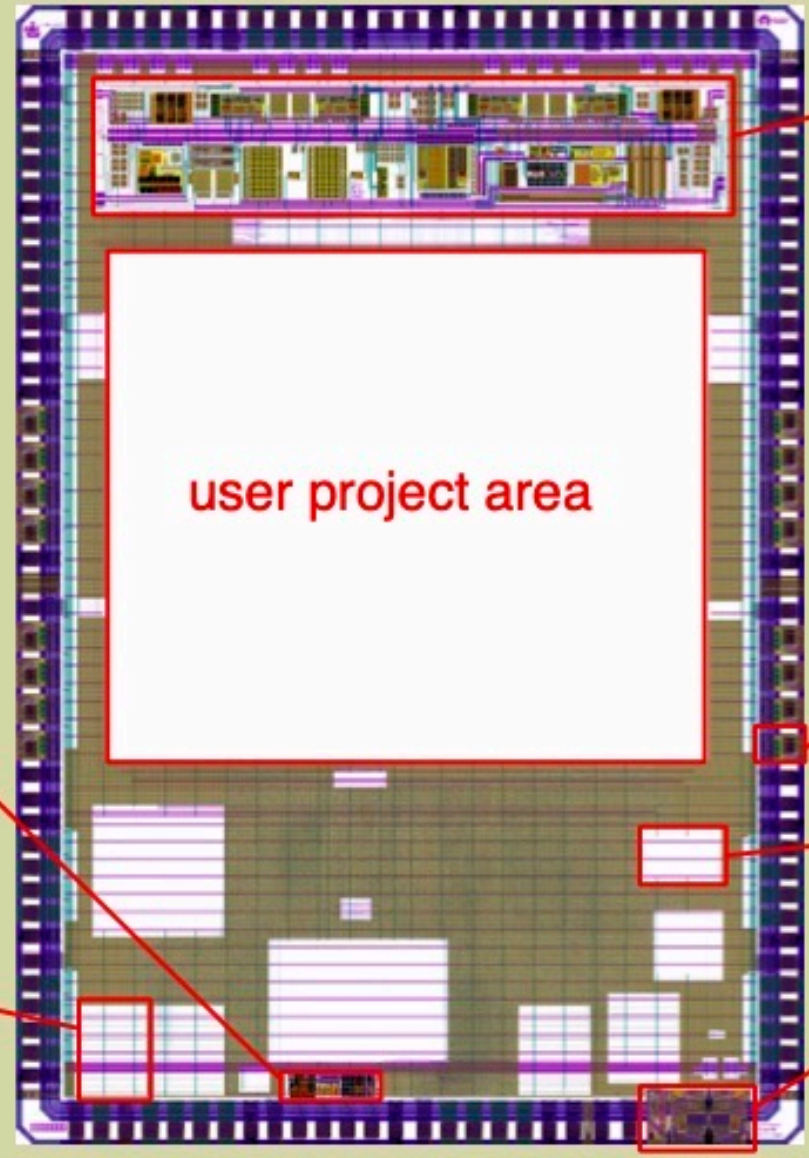
Analog block



https://github.com/fossi-foundation/frigate_analog

“Frigate”: Open Source Mixed-Signal Harness

Frigate top level



OPEN SOURCE PARASITIC EXTRACTION FOR KLAYOUT

FSIC 2025 CONFERENCE

DI Martin Köhler, Senior Software Developer
Univ.-Prof. Dr. Harald Pretl, Head of Institute
Institute for Integrated Circuits
Johannes Kepler University Linz

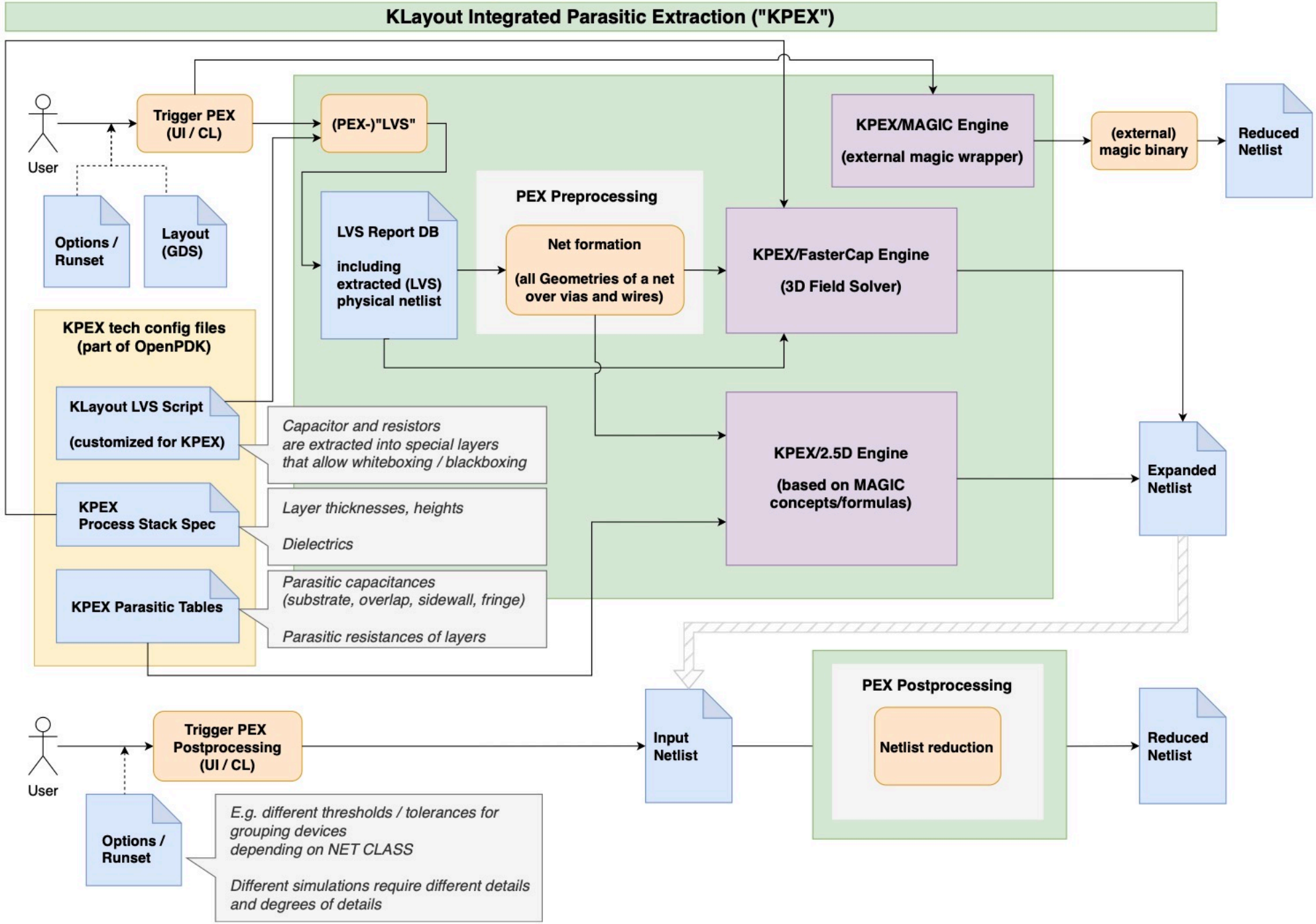
Version 0.9, July 2025

JKU IIC INSTITUTE FOR INTEGRATED CIRCUITS

SAL SILICON AUSTRIA LABS

GETTING STARTED

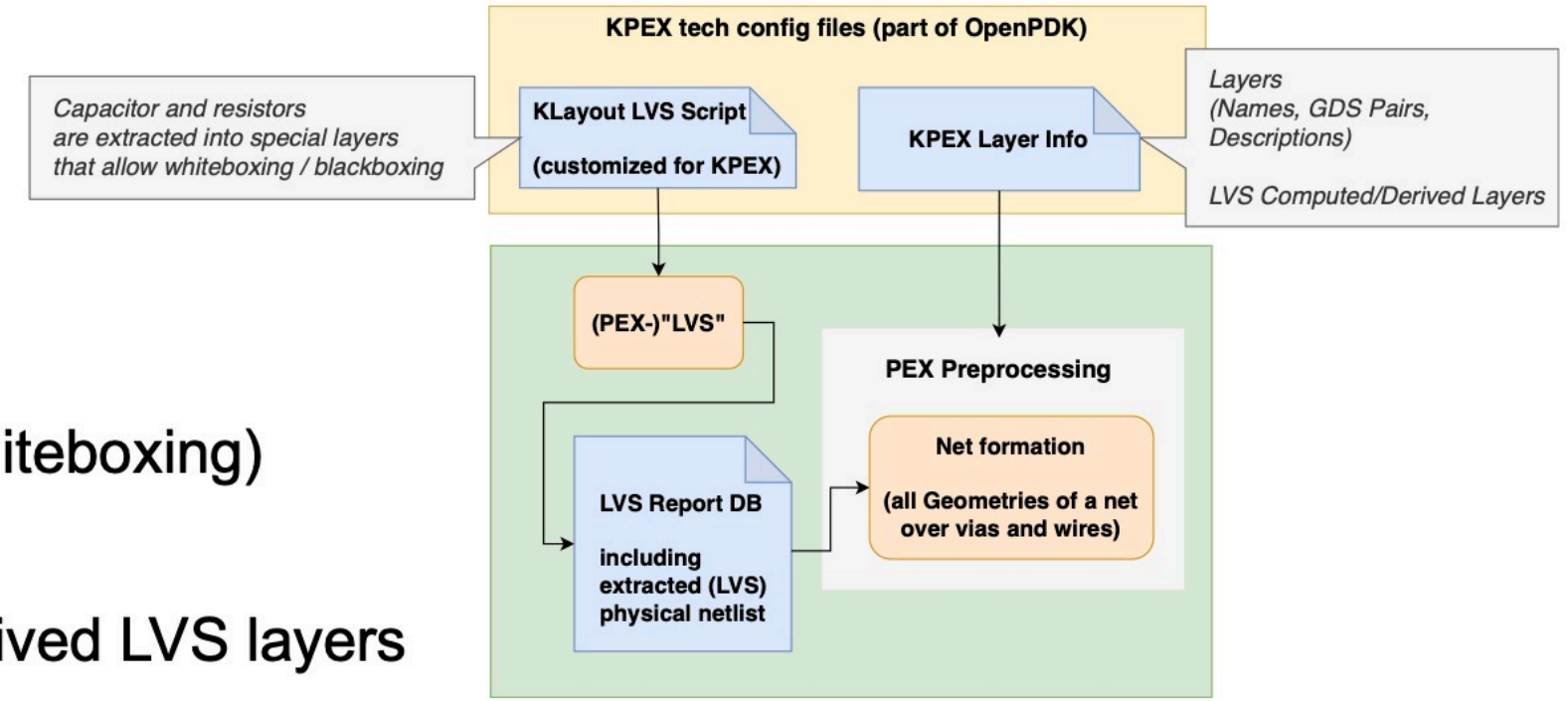
- Repository:
  - https://github.com/martinjankoehler/klayout-pex/
- PyPi:
  - `pip3 install klayout-pex`
  - `kpex --help`
- Documentation:
  - https://martinjankoehler.github.io/klayout-pex-website/doc/doc.html
- Optional:
  - `FasterCap` needs to be in `$PATH`
  - `magic` needs to be in `$PATH`
  - `MeshLab` to preview STL files


KLayout Integrated Parasitic Extraction ("KPEX")
User
Trigger PEX (UI / CL)
(PEX-)"LVS"
KPEX/MAGIC Engine
(external magic wrapper)
(external) magic binary
Reduced Netlist
Options / Runset
Layout (GDS)
LVS Report DB
including extracted (LVS) physical netlist
PEX Preprocessing
Net formation
(all Geometries of a net over vias and wires)
KPEX/FasterCap Engine
(3D Field Solver)
KPEX tech config files (part of OpenPDK)
KLayout LVS Script
(customized for KPEX)
Capacitor and resistors are extracted into special layers that allow whiteboxing / blackboxing
KPEX Process Stack Spec
Layer thicknesses, heights
Dielectrics
KPEX Parasitic Tables
Parasitic capacitances (substrate, overlap, sidewall, fringe)
Parasitic resistances of layers
KPEX/2.5D Engine
(based on MAGIC concepts/formulas)
Expanded Netlist
User
Trigger PEX Postprocessing (UI / CL)
Input Netlist
PEX Postprocessing
Netlist reduction
Reduced Netlist
Options / Runset
E.g. different thresholds / tolerances for grouping devices depending on NET CLASS
Different simulations require different details and degrees of details

NET FORMATION (1/2)

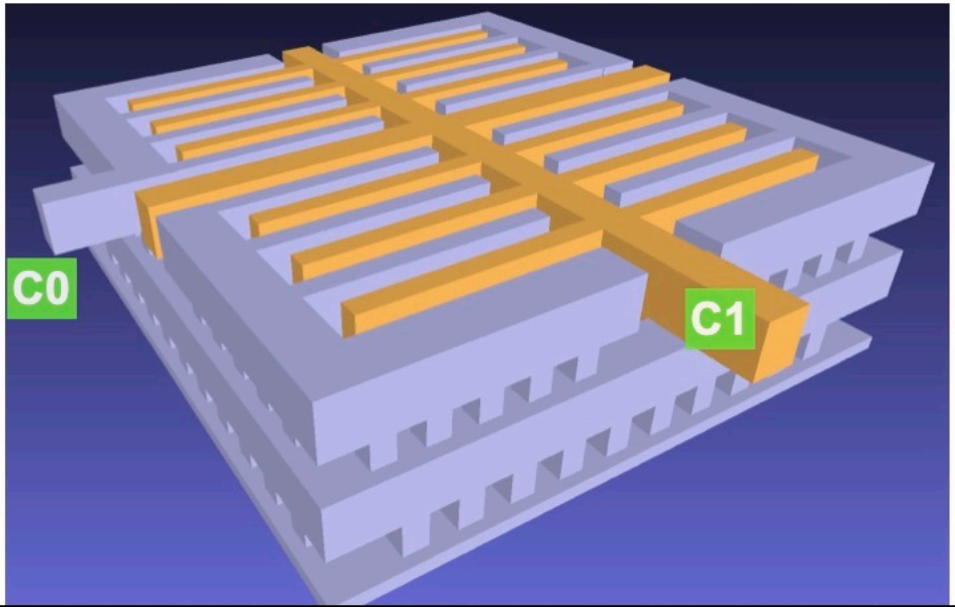
OVERVIEW

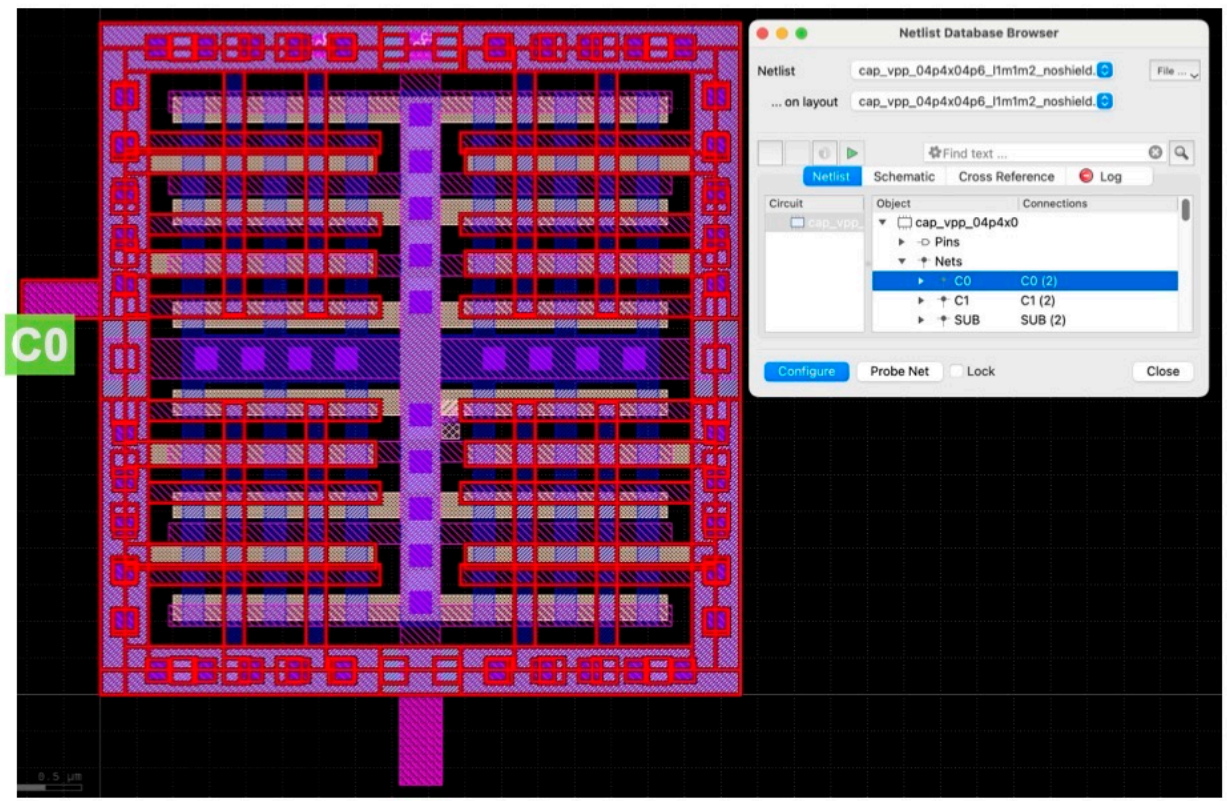
- Custom LVS script
  - layers are named
  - pin layers
  - separate cap layers
    (allows blackboxing / whiteboxing)
- LVS Report DB
  - contains computed / derived LVS layers
  - contains net / connectivity info
- Net formation
  - requires LVSDB and KPEX tech info
  - obtain polygons on each net on each layer

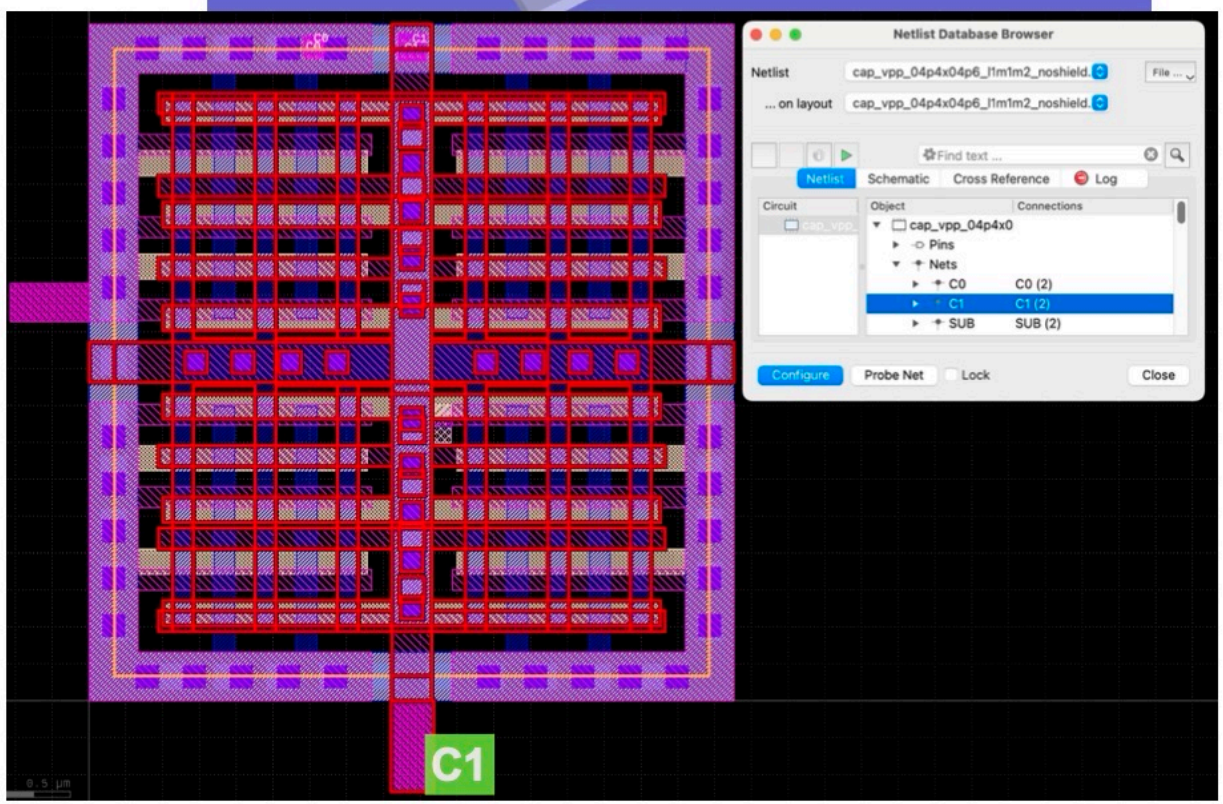


NET FORMATION (2/2)

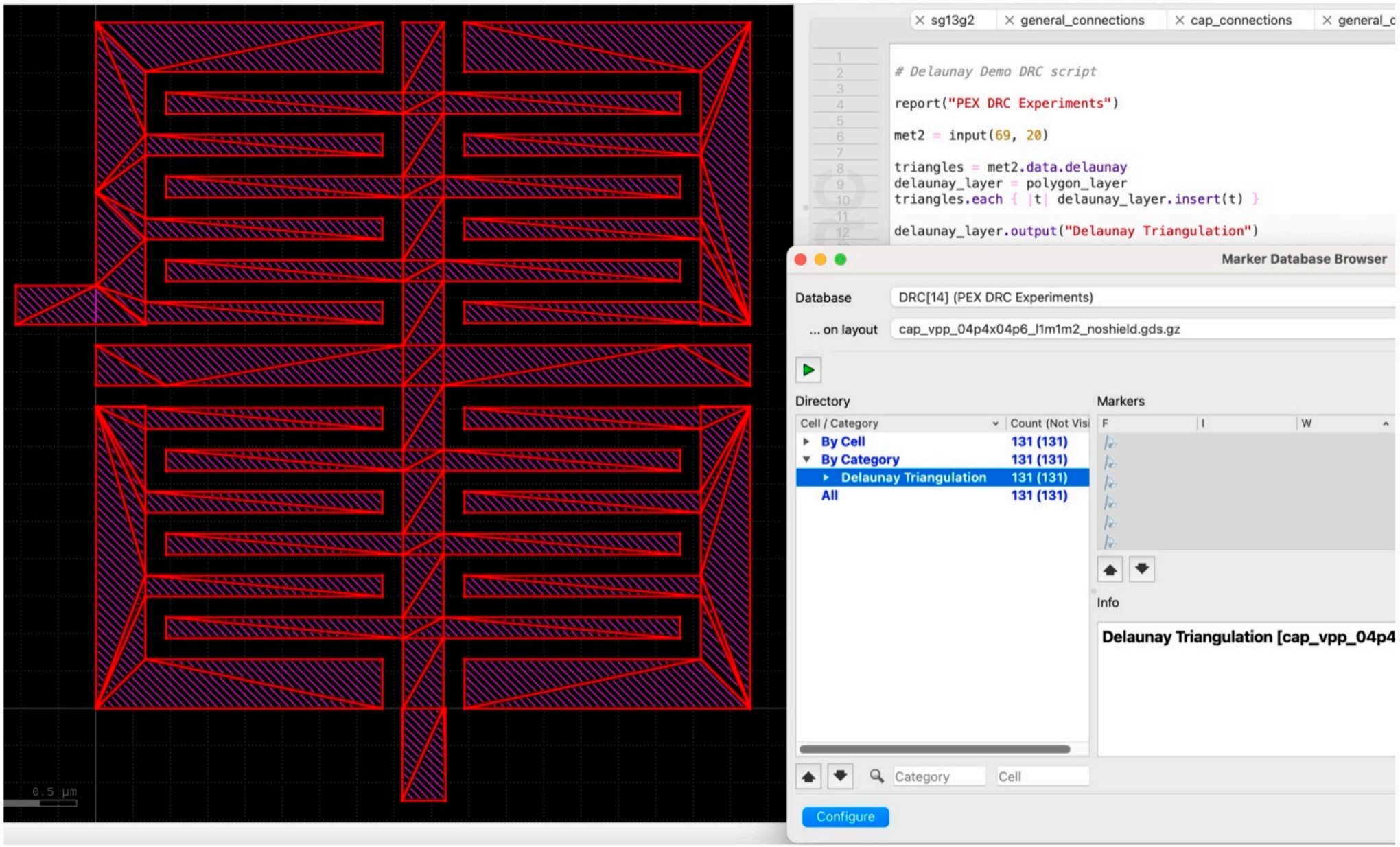
KLAYOUT NETLIST BROWSER





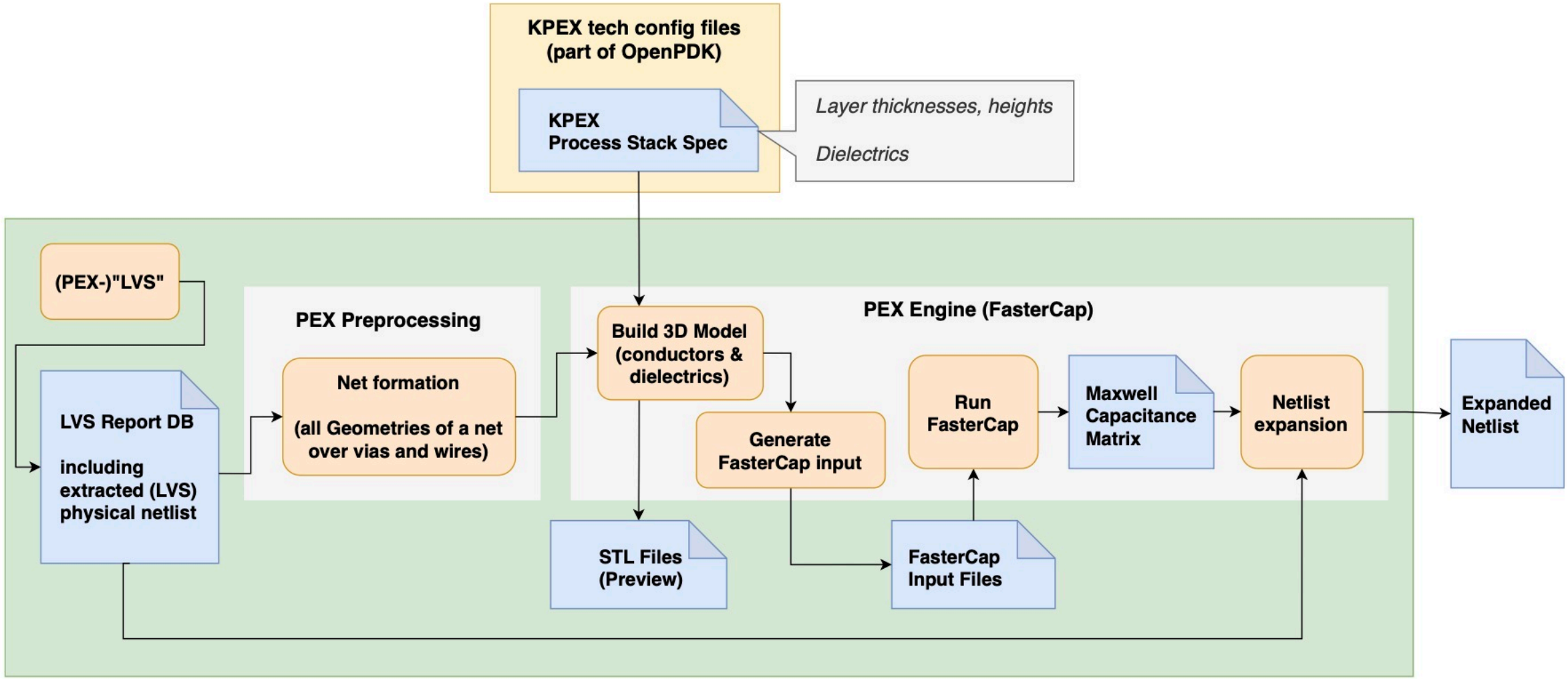


KLAYOUT DELAUNAY TRIANGULATION



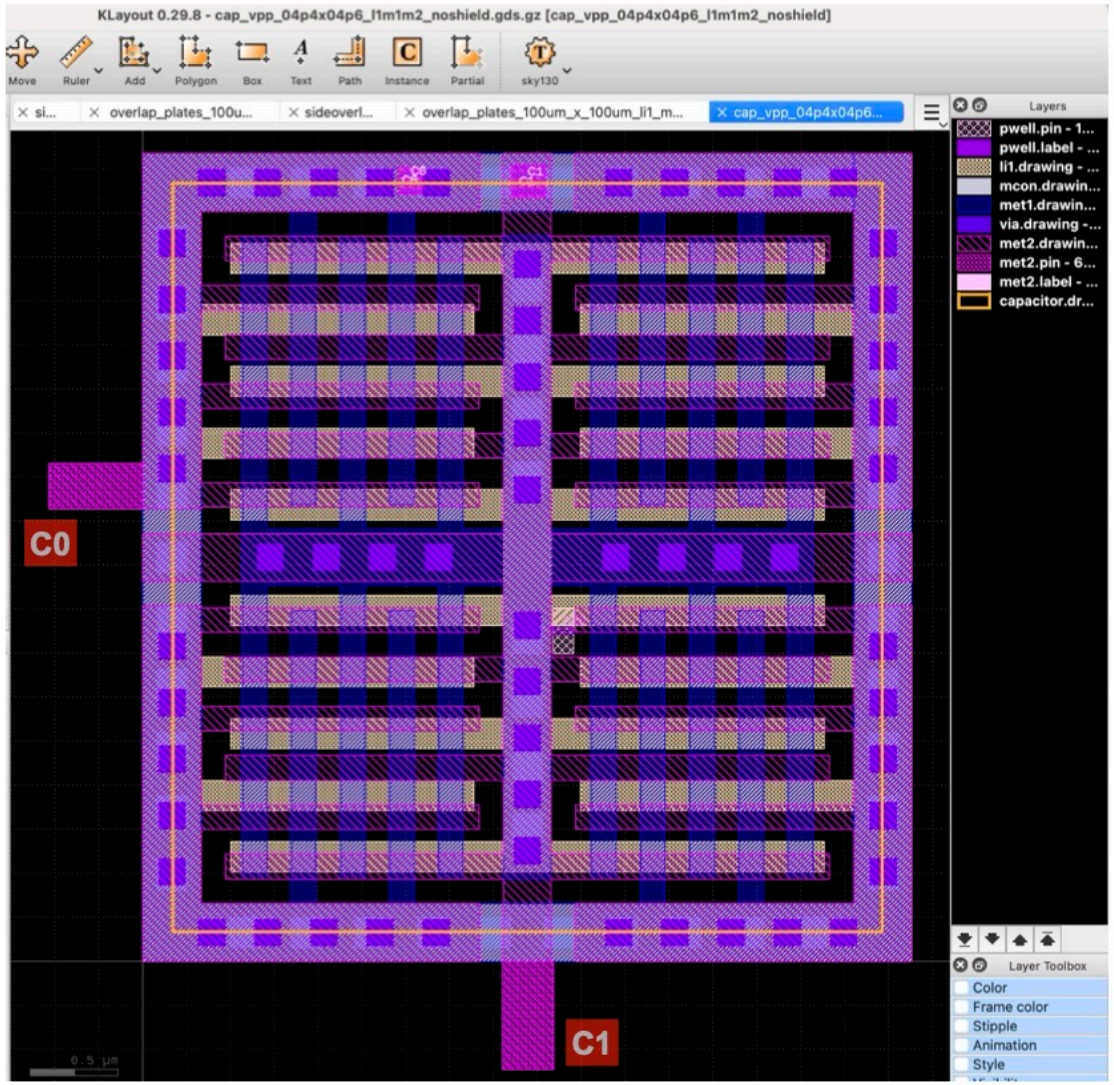
ENGINE: KPEX/FASTERCAP (1/4)

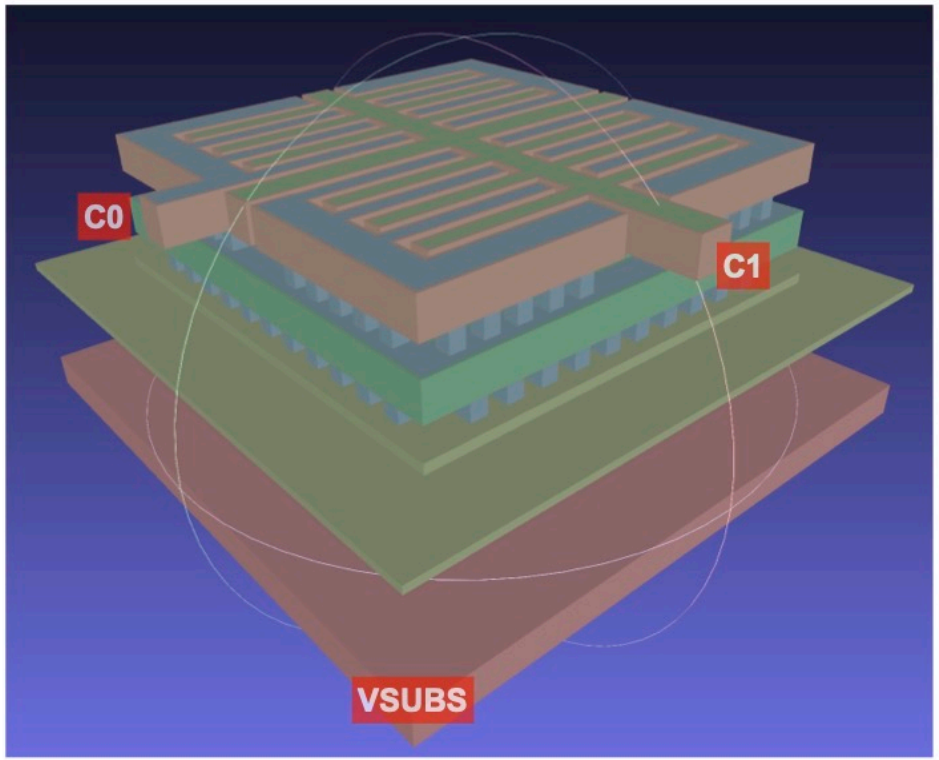
OVERVIEW



ENGINE: KPEX/FASTERCAP (2/4)

3D MODEL





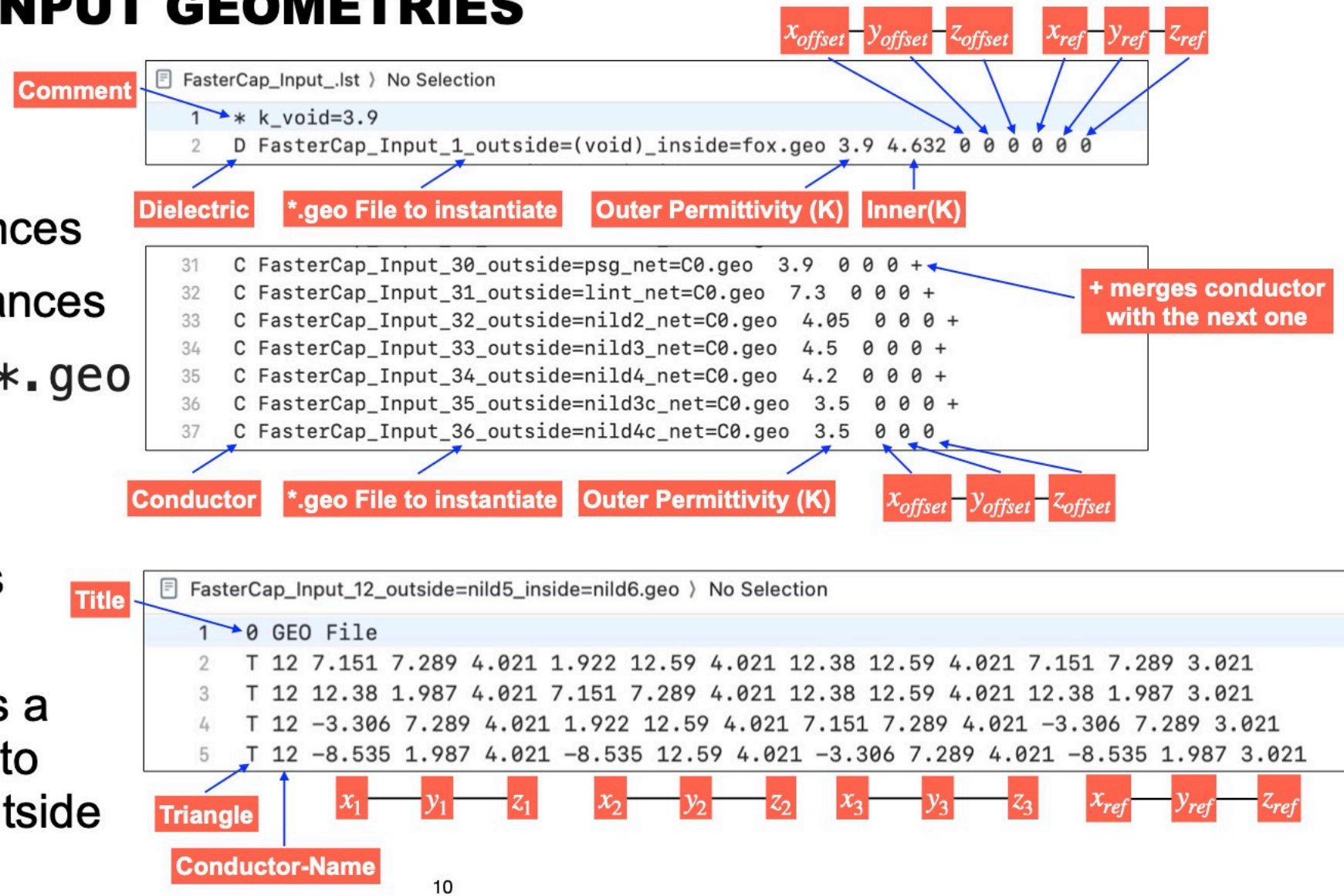
shown without most dielectrics

- build 3D Model
  - using Delaunay Triangulation (2D x/y)
  - using KPEX Tech Info (Process Stackup) for 3D z-Axis Info (Layer z-offset & thickness) and dielectric info
- create STL Files for 3D preview (e.g. using MeshLab)

ENGINE: KPEX/FASTERCAP (3/4)

FASTERCAP 3D INPUT GEOMETRIES

- ∗.lst file
  - Main input file
  - Dielectric Instances
  - Conductor Instances
  - each refer to a ∗.geo file
- ∗.geo files
  - Defines Shapes (e.g. Triangles)
  - Each shape has a reference point to define inside/outside



ENGINE: KPEX/FASTERCAP (4/4)

MAXWELL CAPACITANCE MATRIX

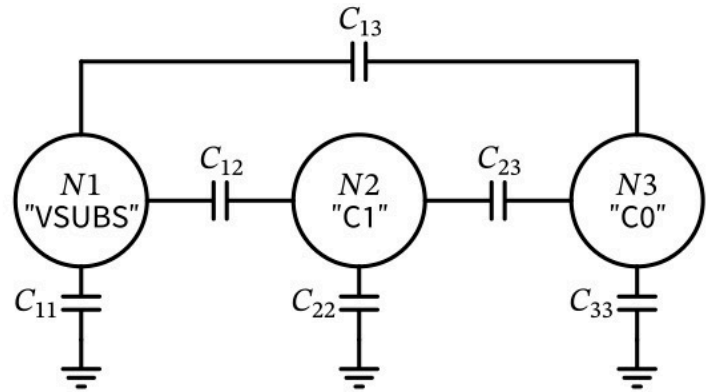
```
Capacitance matrix is:
Dimension 3 x 3
g1_VSUBS  2.08888e-09 -1.46568e-10 -7.37007e-10
g2_C1  -1.64457e-10 1.47687e-08 -1.44368e-08
g3_C0  -7.68811e-10 -1.4528e-08 1.54806e-08

Weighted Frobenius norm of the difference between capacitance (auto option): 0.005031

Solve statistics:
Number of input panels: 18819 of which 5856 conductors and 12963 dielectric
Number of input panels to solver engine: 18819
Number of panels after refinement: 47371
Number of potential estimates: 6249961
Number of links: 19880193 (uncompressed 2244011641, compression ratio is 99.1%)
Max recursion level: 35
Max Mesh relative refinement value: 0.00126534
Time for reading input file: 0.014353s
Time for building super hierarchy: 0.001892s
Time for discretization: 0.562416s
Time for building potential matrix: 0.673007s
Time for precond calculation: 0.251937s
Time for gmres solving: 20.033398s
```

FasterCap Output (units are $/10^{-6}$)

$$C = \begin{matrix} \\ \text{VSUBS} \\ \text{C0} \\ \text{C1} \end{matrix} \begin{matrix} \begin{matrix} \text{VSUBS} & \text{C0} & \text{C1} \end{matrix} \\ \begin{bmatrix} m_{11} - m_{12} - m_{13} & m_{12} & m_{13} \\ m_{21} & m_{22} - m_{21} - m_{23} & m_{23} \\ m_{31} & m_{32} & m_{33} - m_{31} - m_{32} \end{bmatrix} \end{matrix}$$

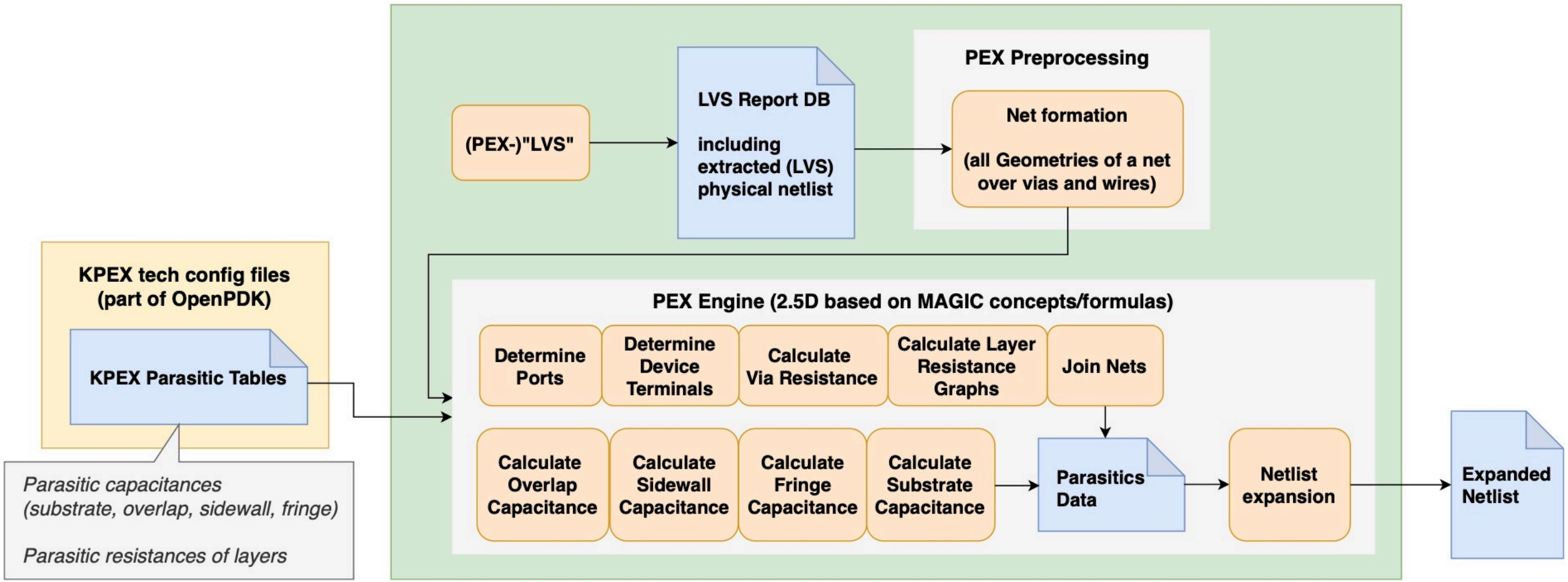


Result CSV

Device	Net1	Net2	Capacitance [F]	Capacitance [fF]
Cext_0_1	VSUBS	C1	1.555125e-16	0.16f
Cext_0_2	VSUBS	C0	7.52909e-16	0.75f
Cext_1_2	C1	C0	1.44824e-14	14.48f
Cext_1_1	C1	VSUBS	1.307875e-16	0.13f
Cext_2_2	C0	VSUBS	2.45291e-16	0.25f

ENGINE: KPEX/2.5D

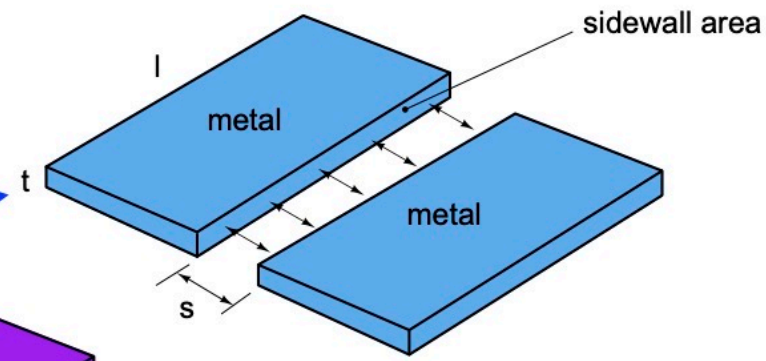
OVERVIEW

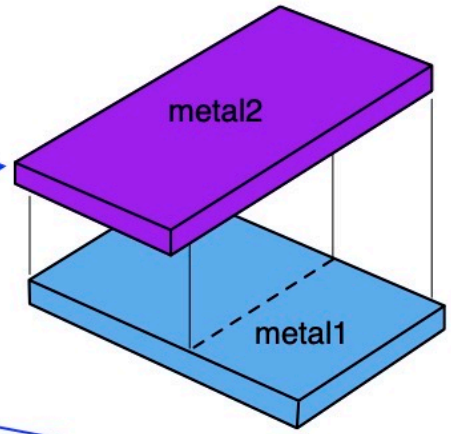


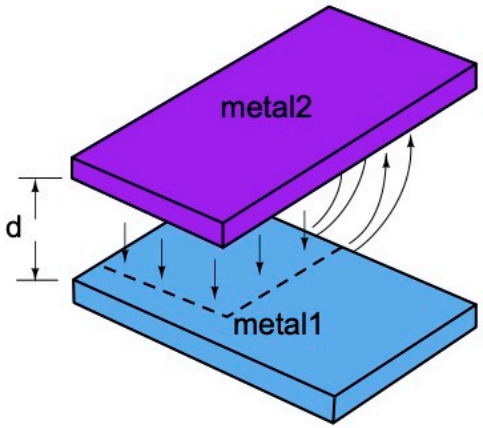
ENGINE: KPEX/2.5D

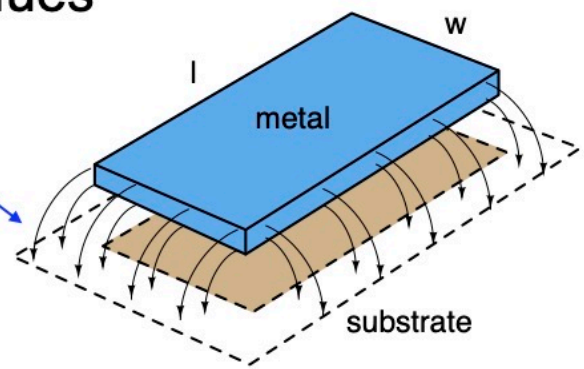
PARASITIC CAPACITANCE TYPES IN MAGIC

- **Sidewall Capacitance**
  - On Same Layer
- **Overlap Capacitance**
  - On different layers
- **Fringe Capacitance ("Side Overlap")**
  - On different layers
  - Both directions, different values
- **Substrate Capacitance**
  - Area
  - Fringe ("Perimeter")





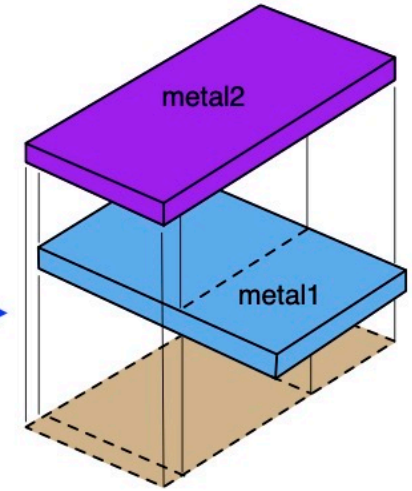


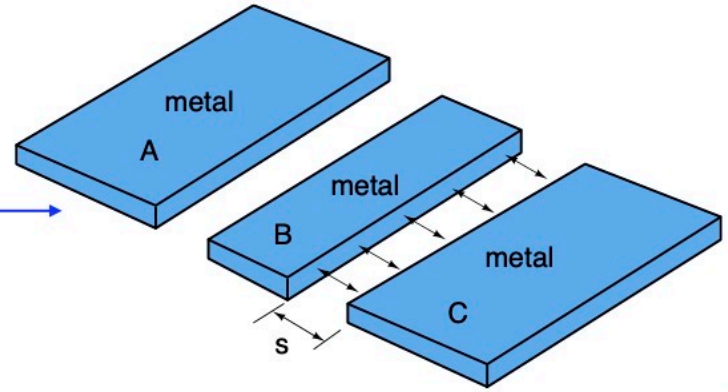


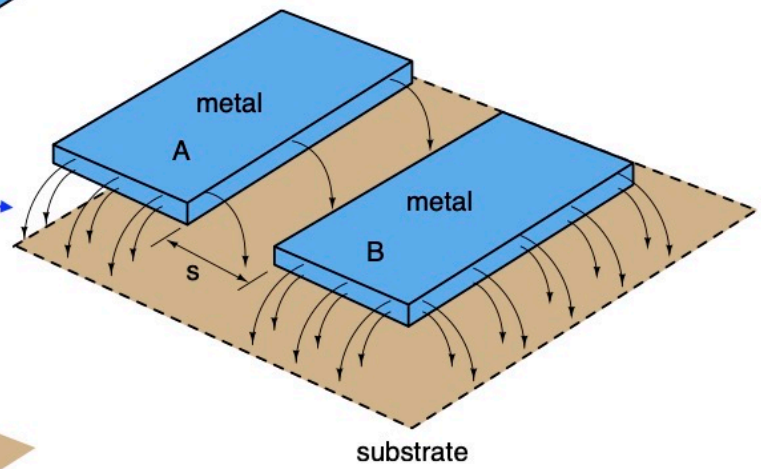
ENGINE: KPEX/2.5D

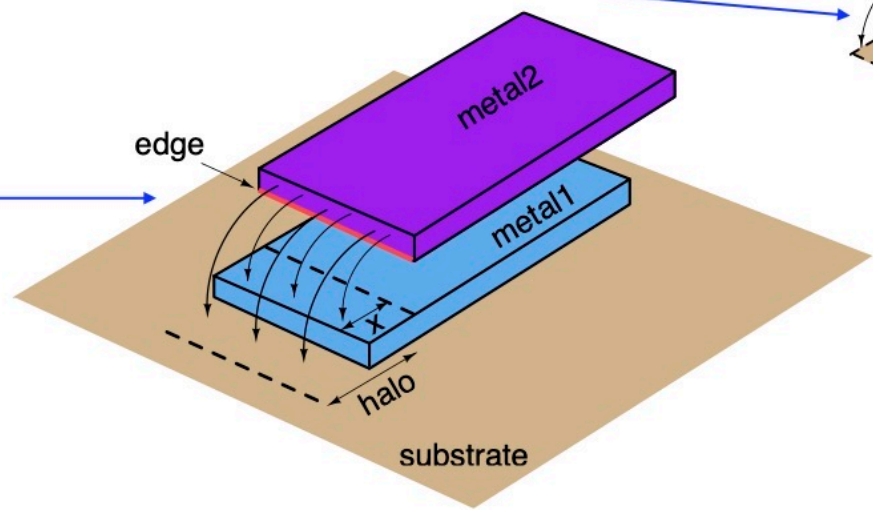
PARASITIC CAPACITANCE – SHIELDING

- **Overlap Shielding**
  - ☐ To Substrate
  - ☐ Between Metal Layers
- **Lateral Shielding**
  - ☐ On Same Layer
- **Lateral Fringe Shielding**
  - ☐ On Same Layer
  - ☐ Between Layers
- **Vertical Fringe Shielding**
  - ☐ To Substrate
  - ☐ Between Layers



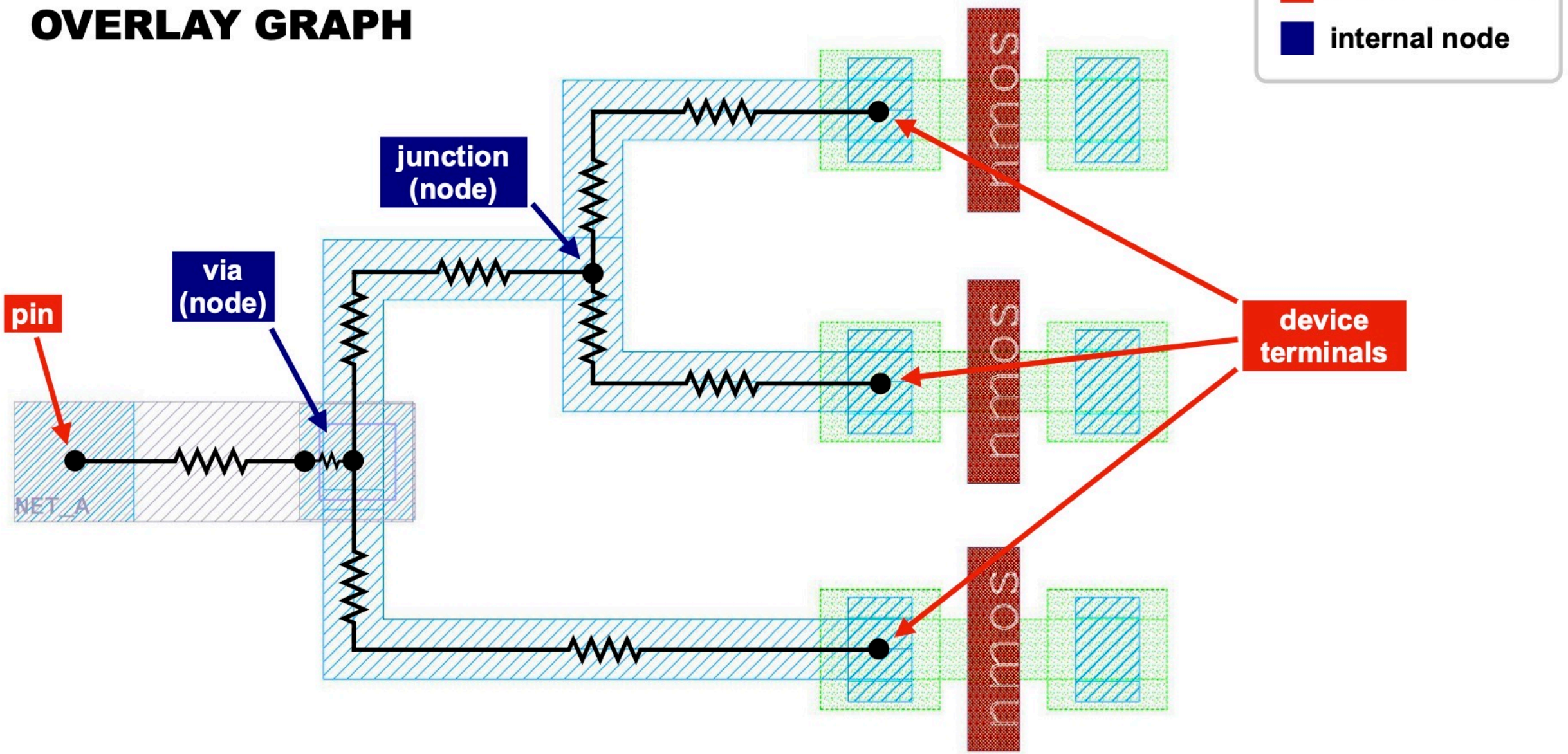






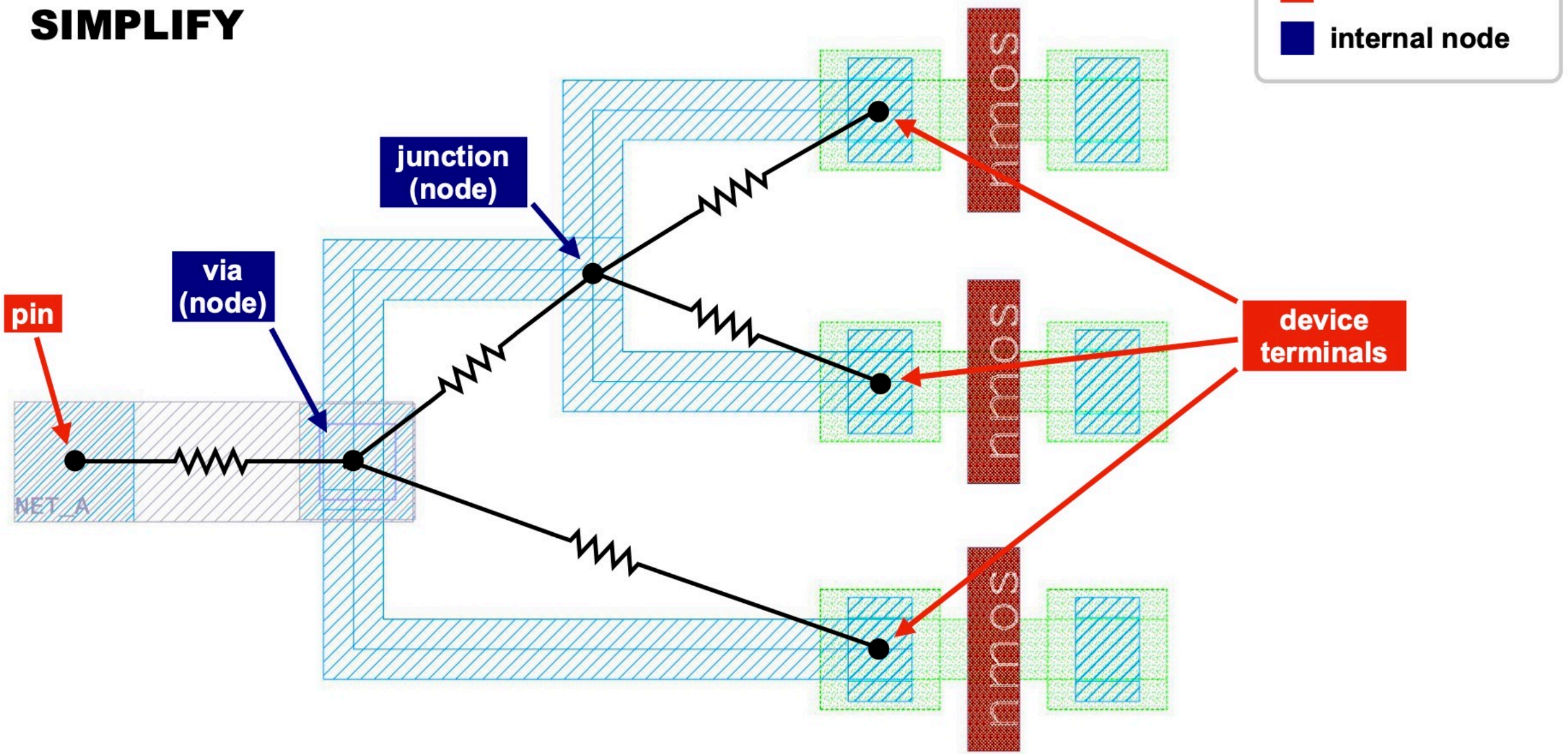
ENGINE: KPEX/2.5D

OVERLAY GRAPH



ENGINE: KPEX/2.5D

SIMPLIFY



STATUS: DONE

- **Understand KLayout API**
- **Net Formation using KLayout API**
  - ☐ Customized PEX-LVS scripts
- **Technology File**
  - ☐ Format (Layers, LVS Layers, 3D Stack-up, Parasitic Tables)
  - ☐ Technology File for Sky130A
  - ☐ Technology File for IHP SG13G2
  - ☐ Validate Technology File feasibility using KPEX/FasterCap (3D) and KPEX/2.5D (analytic, MAGIC style)
- **KPEX CLI Tool**
- **Post-Processing flow phase (independent of PEX engine)**
  - ☐ Trivial netlist reduction (parallel C, serial R)

STATUS: DONE

- **KPEX Engine FasterCap Integration**
  - ☐ 3D Geometry Generation using KLayout API
  - ☐ Generate FasterCap Input Files (and STL meshes for preview)
  - ☐ Interpret Maxwell Capacitance Matrix
- **KPEX Engine MAGIC Integration** *(external binary)*
- **Feasibility Study: Netlist Post-Processing Reduction**
  - ☐ Review Time Constant Equilibration Reduction (TICER 4)

STATUS: DONE

- **KPEX/2.5D Engine based on MAGIC (CC)** *(partially DONE, due to additional tasks)*
  - ☐ Extract knowledge about MAGIC capacitance algorithms
  - ☐ Understand MAGIC CC formulas (overlap, sidewall, fringe)
  - ☐ Find and apply KLayout API means to help implement these formulas (geometric, LVS/DRC methods, …)
  - ☐ Implementation of overlap capacitance (including shielding)
  - ☐ Find means to visualize/debug contributions using KLayout Marker Database
  - ☐ Implementation of sidewall capacitance (including shielding)
  - ☐ Implementation of fringe capacitance (and shielding)
  - ☐ [Verification] Compare results with MAGIC
  - ☐ Debugging
  - ☐ [Verification] build suite of (unit) test layouts
  - ☐ [Verification] regression test suite

STATUS: IN PROGRESS / NEXT STEPS

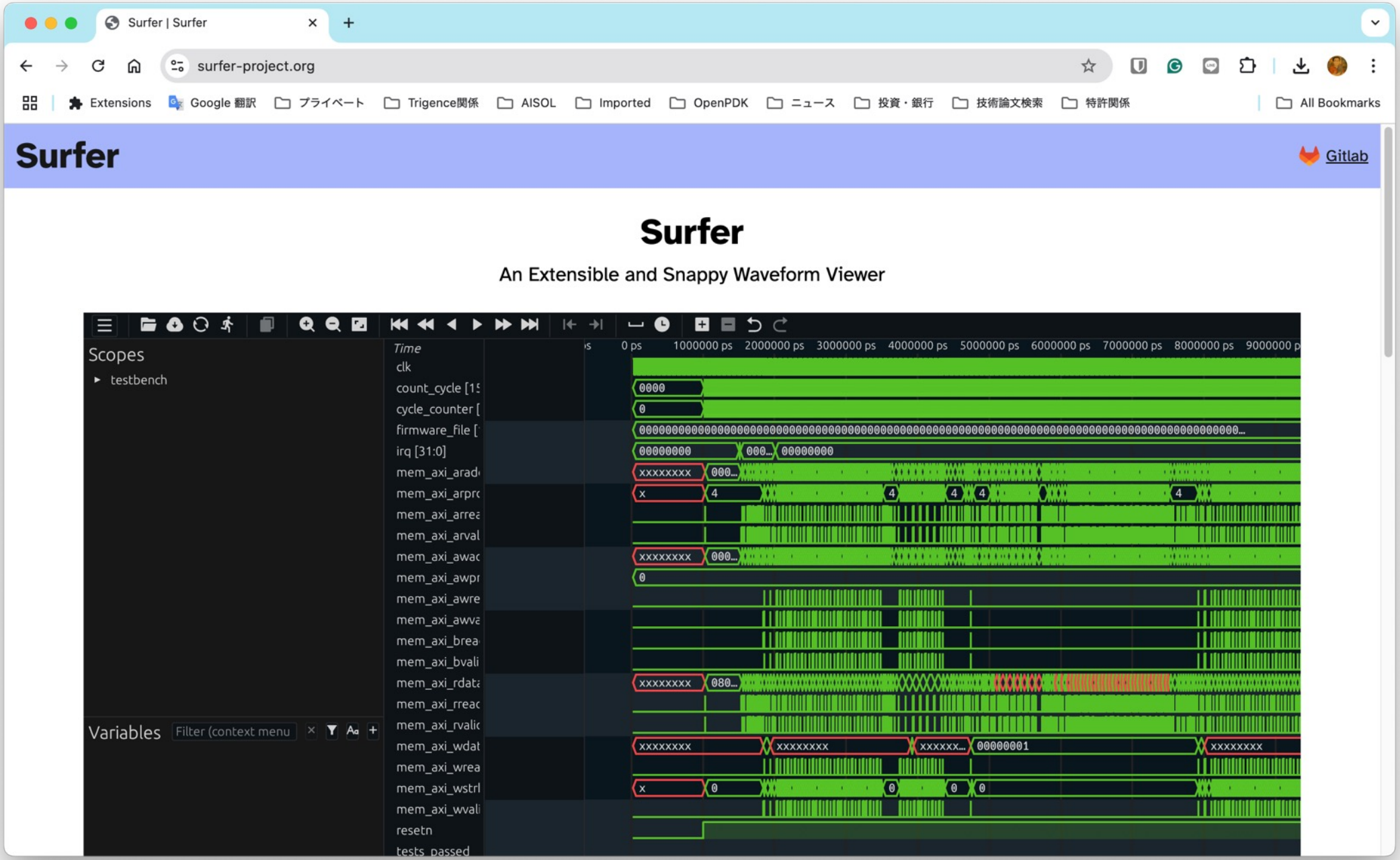
- **KPEX Engine based on MAGIC (RC)**
  - ☐ Extract knowledge about MAGIC resistance algorithms
  - ☐ Understand MAGIC formulas (R)
  - ☐ Find KLayout means to extract a node graph from the layout
  - ☐ Implement RC Mode
  - ☐ [Verification] Compare results with MAGIC
  - ☐ [Verification] regression test suite

- **Blackboxing / Whiteboxing for Devices** *(e.g. MoM / MiM Caps with SPICE models)*

- **Validation using larger layouts** *(currently smaller layout examples)*

STATUS: FUTURE WORK

- **KPEX/2.5D Analytical Engine**
  - Hierarchical Optimization *(currently flattened to a single cell)*
  - GUI or Run-sets not yet implemented *(currently lassic CLI Tool with args)*
  - C++ Port / Optimization *(currently Python prototype, using same C++ API)*
- **Process Corners** *(could currently be mimicked by providing multiple tech files)*
- **Comparison MAGIC vs FastHenry/FasterCap vs KPEX/2.5D**
- **KPEX Engine based on FastHenry**
  - 3D Geometry Generation using KLayout API
  - Comparison with MAGIC and KPEX/2.5



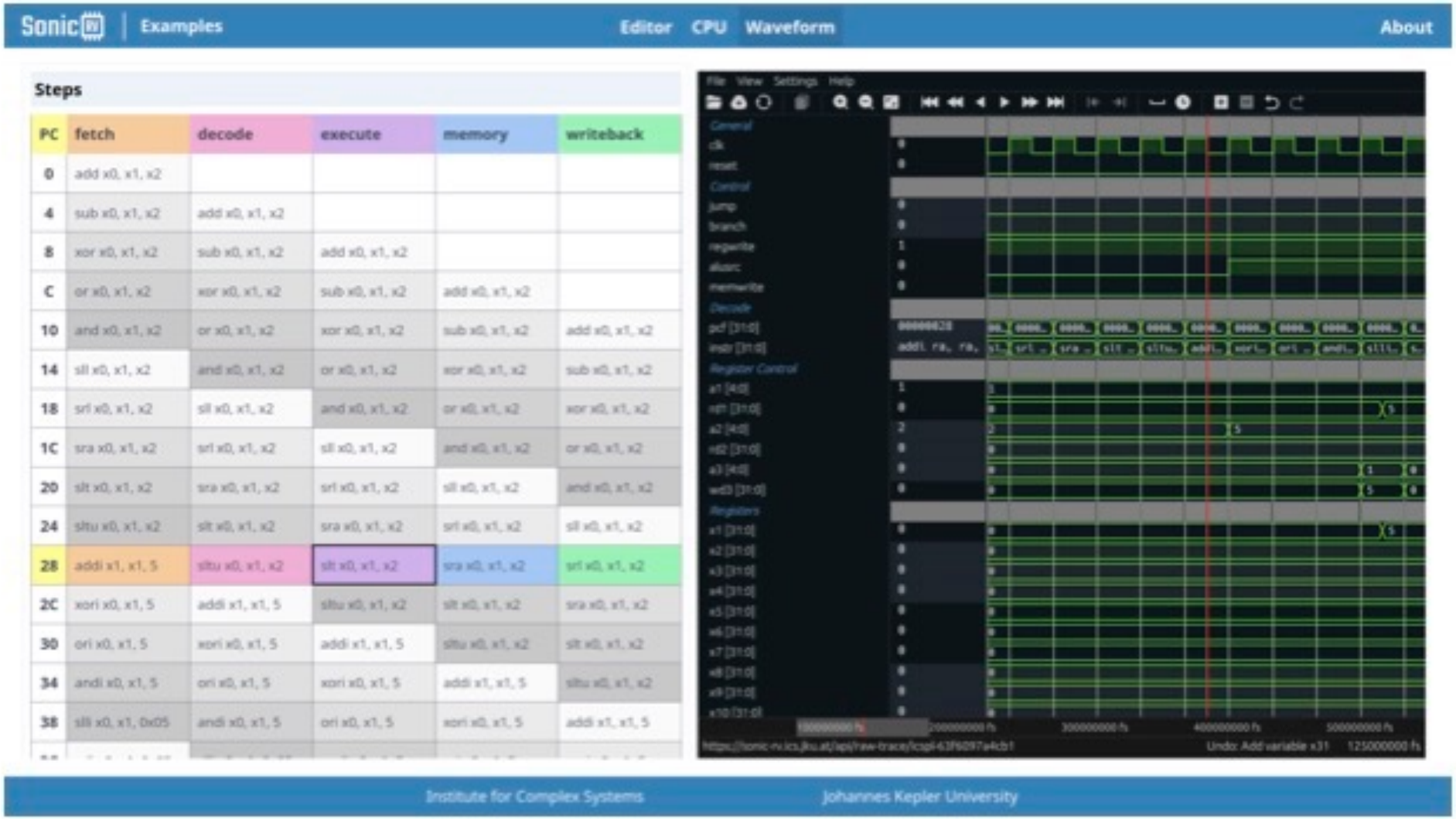
New in Surfer 0.4.0

- **Translator plugins**
- **Signal grouping**
- **Waveform Control Protocol** improvements
- **130** new MRs!

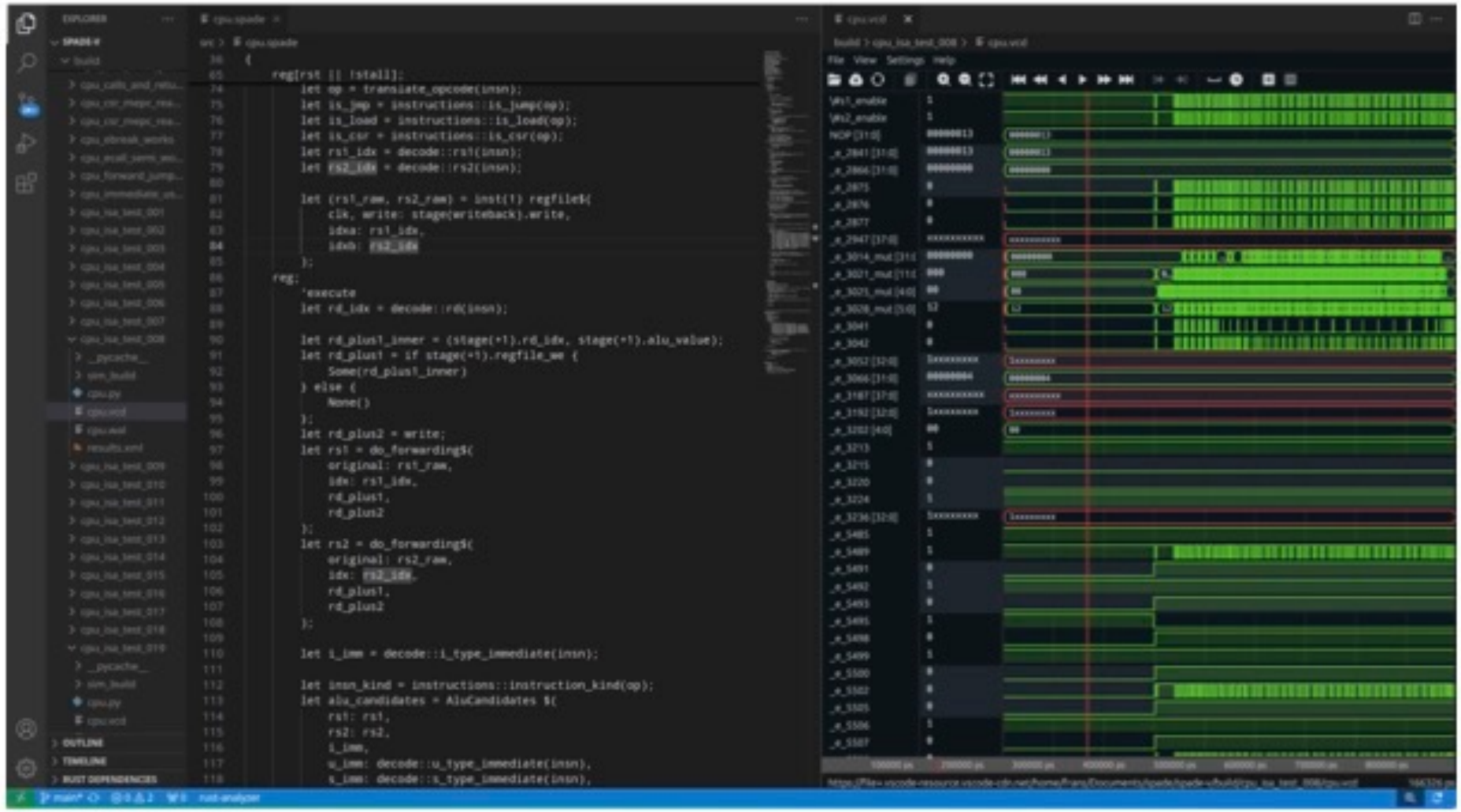
Web Assembly!

- No installation
  - Inspect waves in continuous integration
- **Embeddable**

Embeddable as a teaching tool



Embeddable as a VSCode plugin



Conclusions

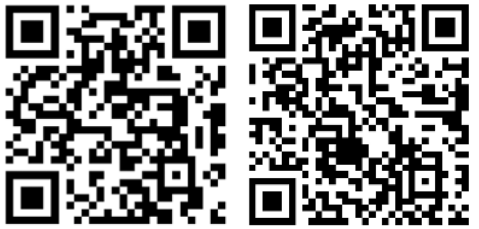
Surfer is a **snappy** and **extensible** waveform viewer that **runs everywhere**

What do **you** want from a waveform viewer?

Try it on your phone!

HM

LINKÖPING UNIVERSITY



Website: ysyx.org/en Telegram Group
Email: ysyx@bosc.ac.cn

"One Student One Chip" Initiative

Learn to Build RISC-V Chips from Scratch with MOOC

OSOC Team

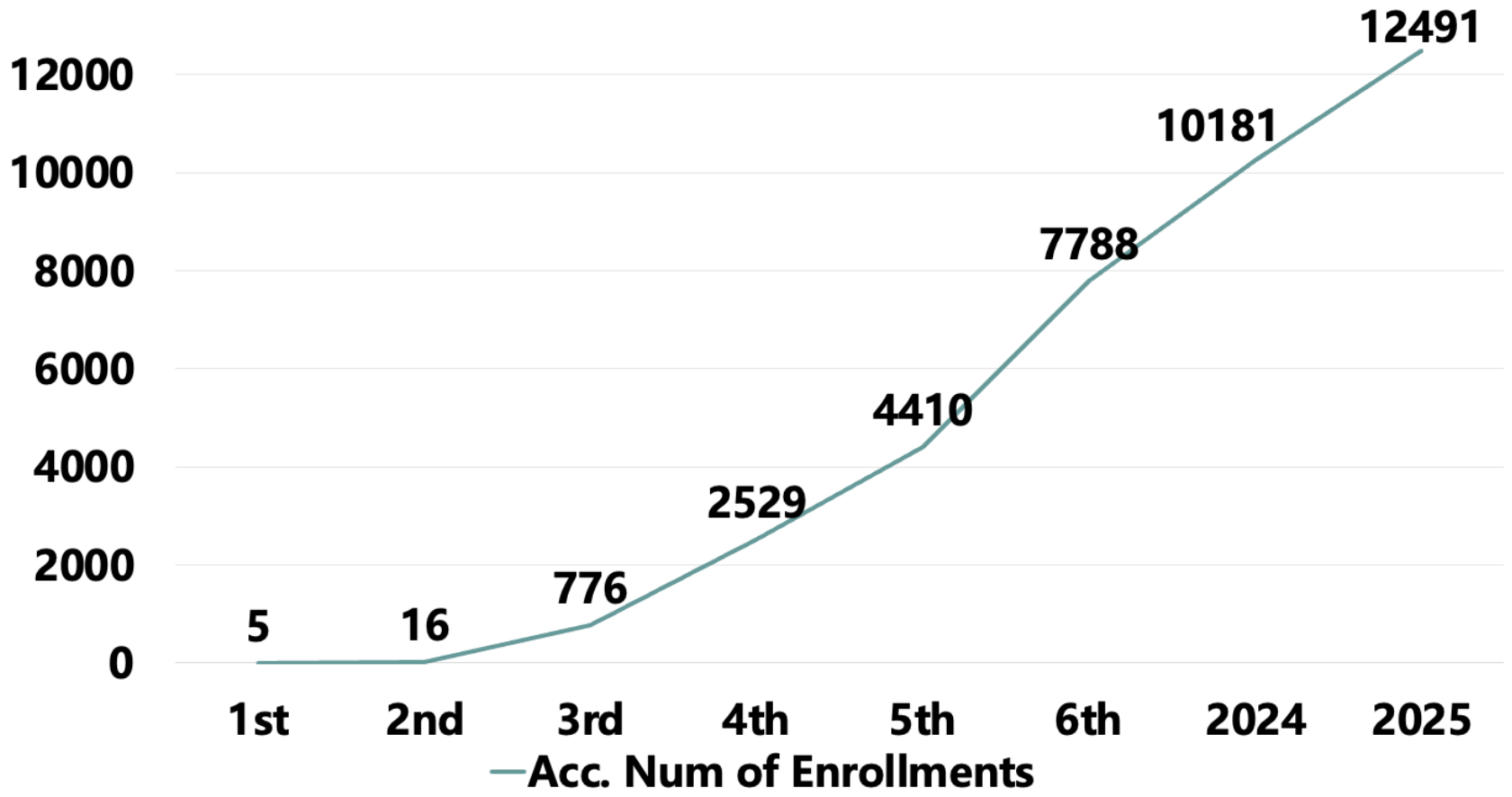
2025.07

中国科学院大学
University of Chinese Academy of Sciences

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

北京开源芯片研究院
BEIJING INSTITUTE OF OPEN SOURCE CHIP

Explore Large-Scale (from 2021-)



	1st	2nd	3rd	4th	5th	6th	2024	2025
Acc. Enrolls	5	16	776	2529	4410	7788	10181	12491

Update on:2025/06/29

OSOC Course Design

Based on open-sourced, practice-oriented, open learning

Everyone is welcomed.
NO limitation
on

- university
- major
- grade
- basis

SW & HW co-design

Logical design & Physical design co-ordination

Application
Runtime
(Simple) OS
ISA(RISC-V)
Micro-architecture
Circuit
Synthesis
Physical design
Physical verification
GDSII

CS
EE

GCC U-boot OpenSBI
UEFI **Software** LLVM
QEMU Linux

XiangShan Coherency
IP OoO
Chip Prefetch
Cache Branch
Extension Prediction

Technology mapping
Place Standard cell
EDA Floorplan
Route
Timing analysis
Clock tree Equivalence

Education equality

Full-stack training

Enter community or company

Learning Materials are opened

The 6th "One Student One Chip" Program Home Page

- Time: Every Saturday 15:00~17:00 China Standard Time
  - Bilibil Live | recording

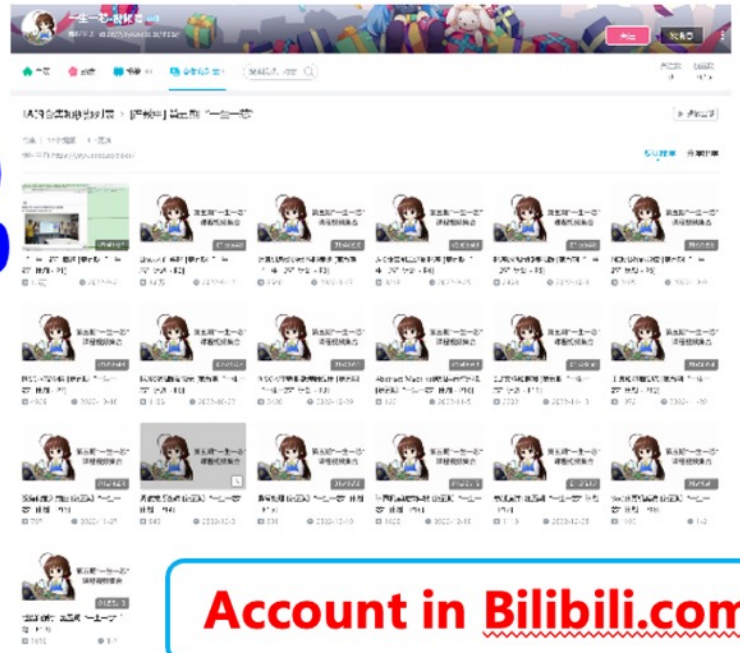
Learning Objectives

"One Student One Chip" will develop your general skills. At the end of the course, you will have a better understanding of the following questions:

1. how processors are designed?
2. how programs run on computers?
3. how to optimize the performance of a processor?
4. how to use/design the right tools for efficient debugging?
5. how to write your own test cases for unit testing?
6. how does an RTL design generate a flowable layout?

We will guide you to design a RISC-V pipeline processor. Run an operating system on your processor. Run a real game on the OS. The processor that achieves the target will be connected to the SoC and will be given the opportunity to tapeout.

- **Course Website**
- **Handouts(260,000 words)**
- **Slides(>800 pages, 85,000 words)**
- **Videos(> 40 hours)**



Account in Bilibili.com

Linux system installation and basic usage

Install a Linux operating system

We're going to reuse the contents of the PA handout, and we're going to ask you to follow PA0 to install Linux OS.

Get "One Student One Chip" code framework

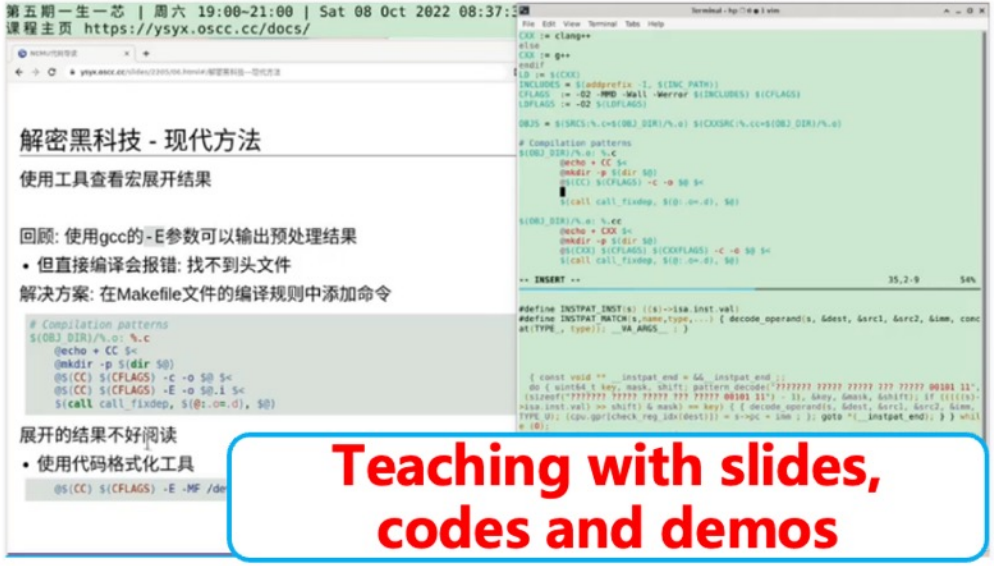
When you read the PA0 handout and proceed to the section on getting the PA framework code, you will be prompted with a box asking you to return to the content of the handout here.

First of all, please add a ssh key on github, please STFW on how to do that. Then get the framework code of "One Student One Chip" by the following command:

```
git clone -b master git@github.com:OSCPU/ysyx-workbench.git
```

Once you have it, you can go back to the appropriate place in the PA handout and continue reading. But you should also note that:

- Please use `ysyx-workbench` as the project directory in the PA handout, i.e. replace occurance of `ics2022` in the PA handout with `ysyx-workbench`.
- When change the student number and name in `ysyx-workbench/Makefile`, you can leave the student number unchanged until you have completed the preliminary.

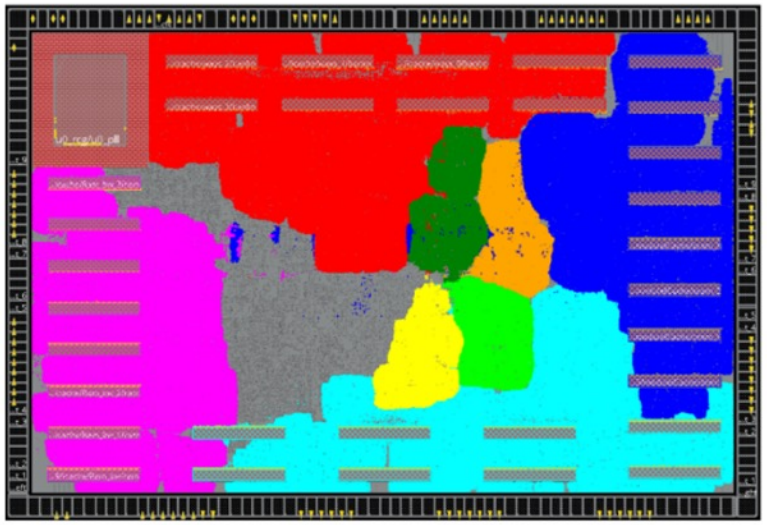


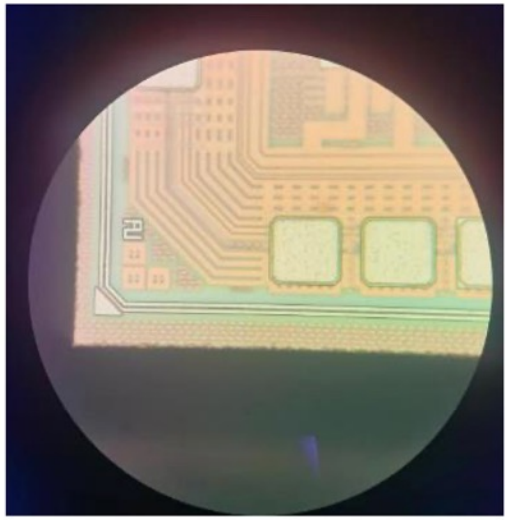
Teaching with slides, codes and demos

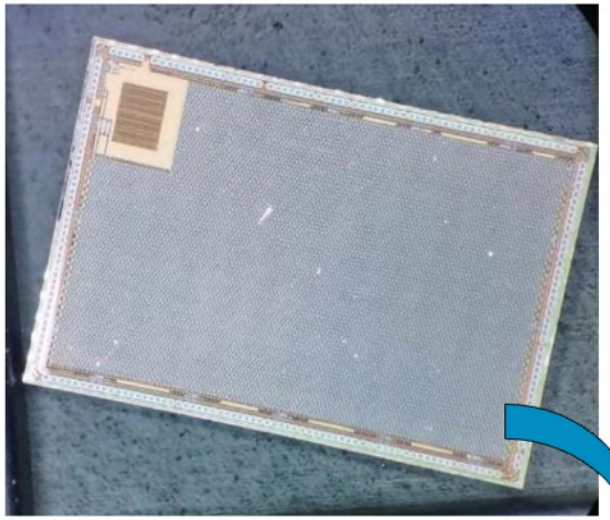
OSOC Roadshow Event

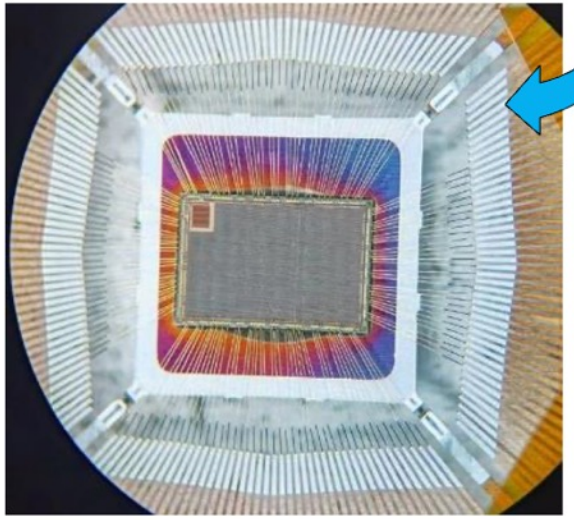
- **Our journey is ongoing, we warmly invite you to reach out and extend an invitation to OSOC to visit your location!**

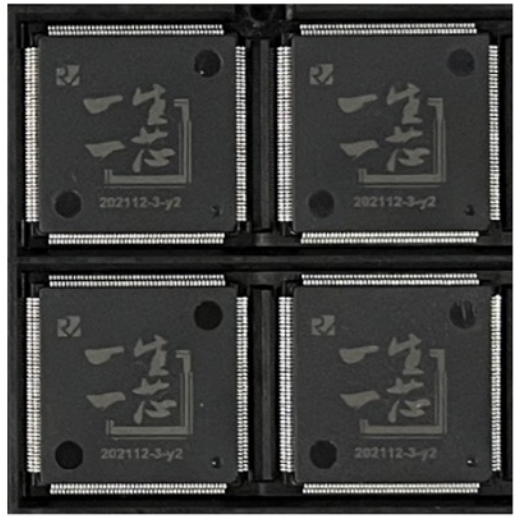
Chip & Board

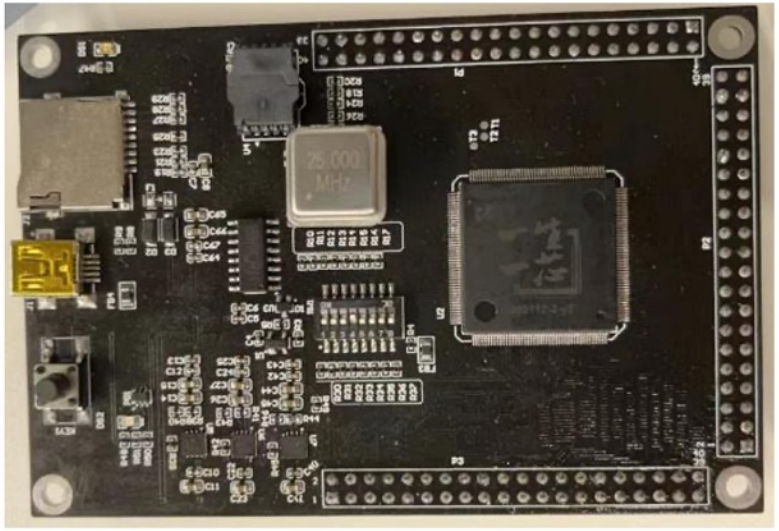






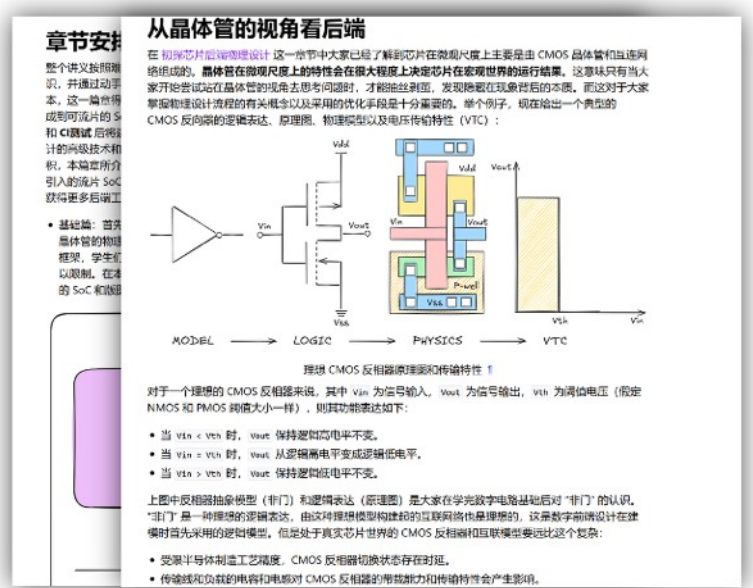


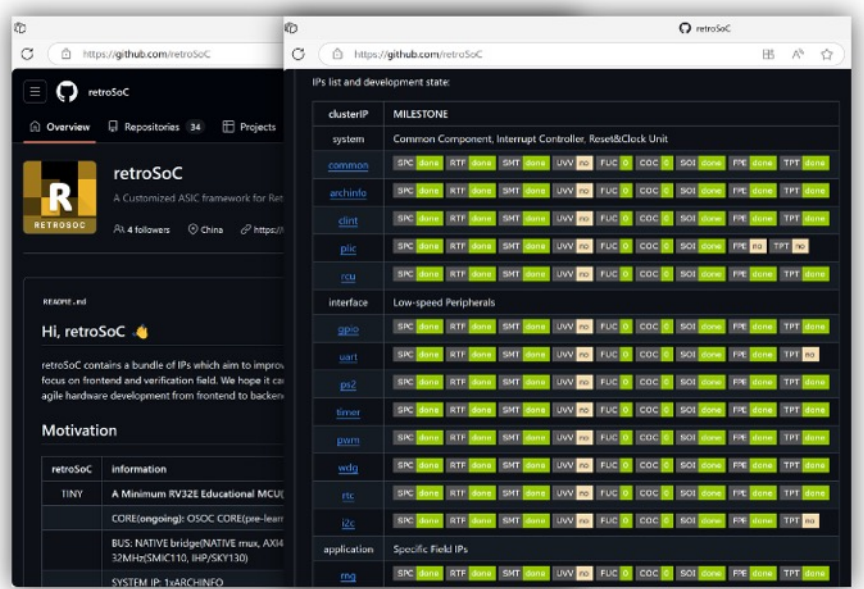


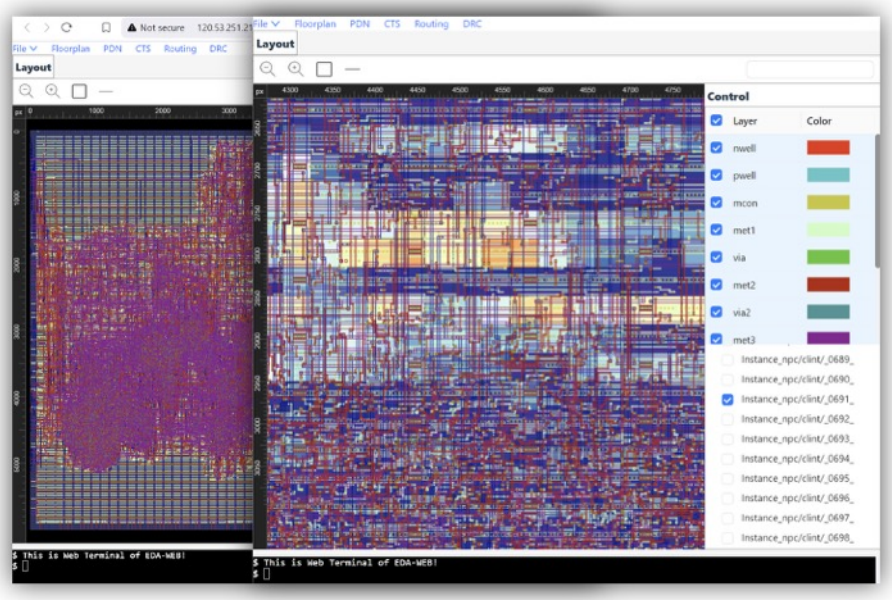


Highlight

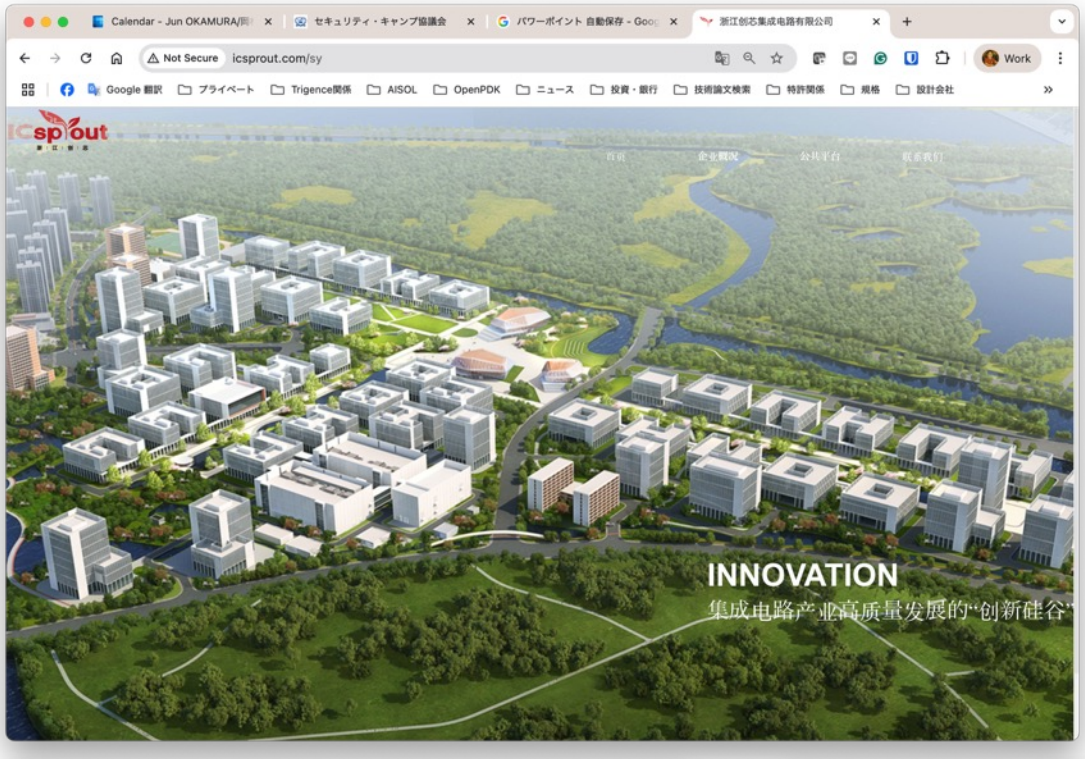
- Teach physical design with cloud platform ***ECOS Studio*** **(Yosys + iEDA + ICS55)**
- Introduce an education-oriented chip ***retroSoC*** tape-outed in **SMIC/SKY130**
- Share some of the latest information on iEDA toolchains

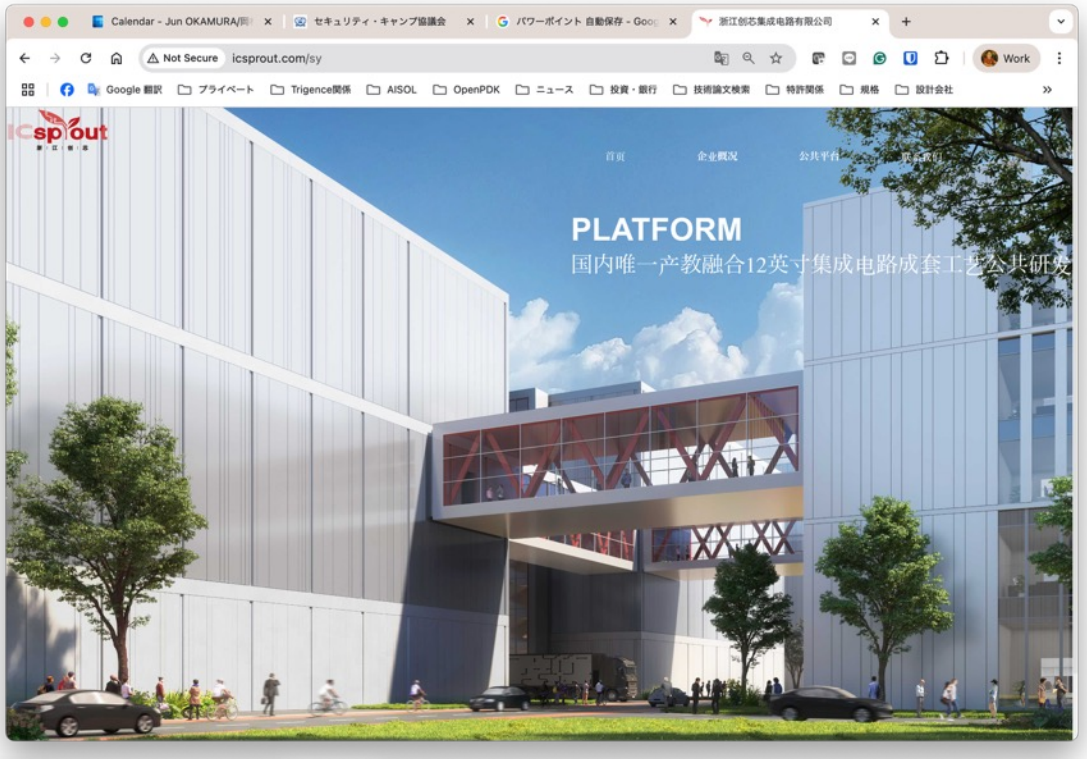


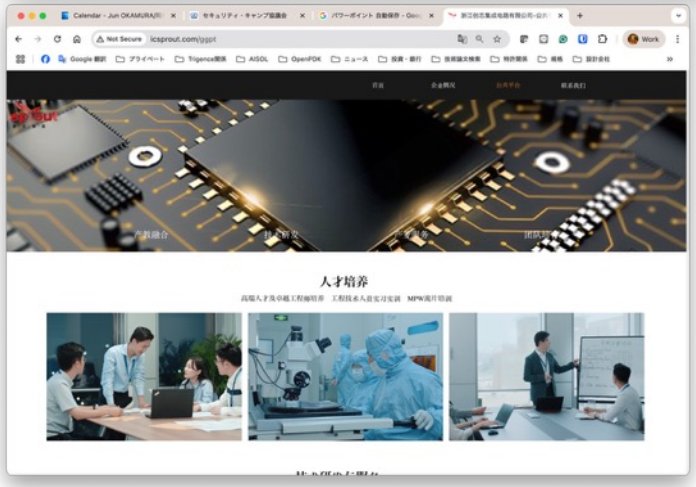


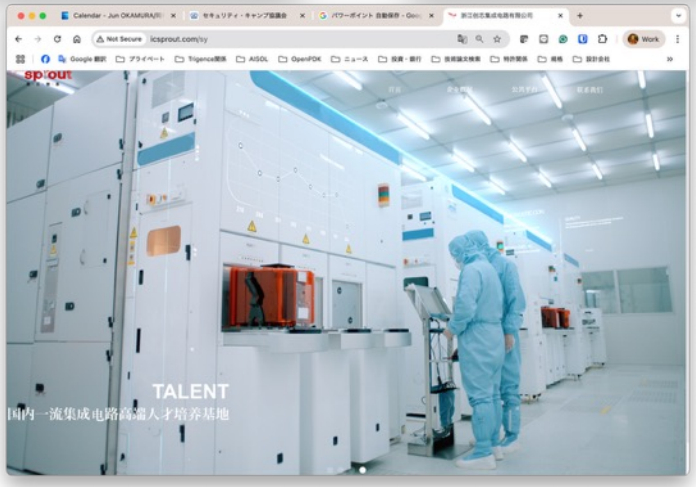


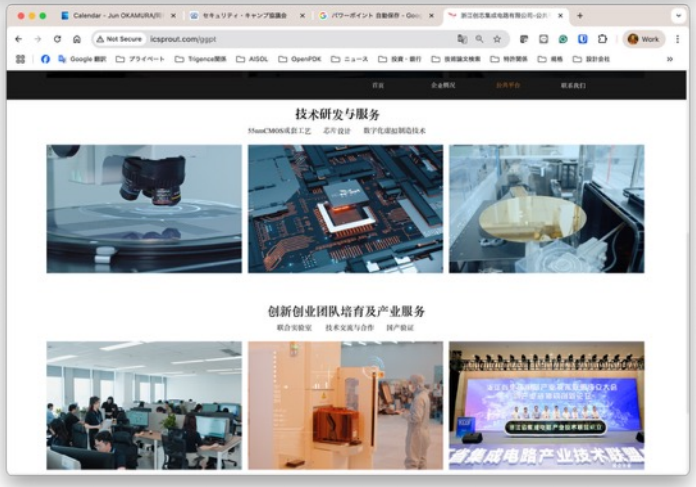
浙江创芯集成电路有限公司(55nm/300mm)











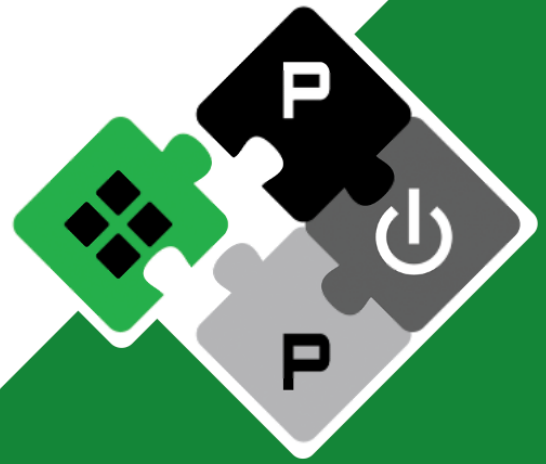
ETH zürich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

End-to-end open-source IC design

Digital Circuits and Systems Group and the PULP team

Frank K. Gürkaynak kgf@iis.ee.ethz.ch



pulp-platform.org

@pulp_platform

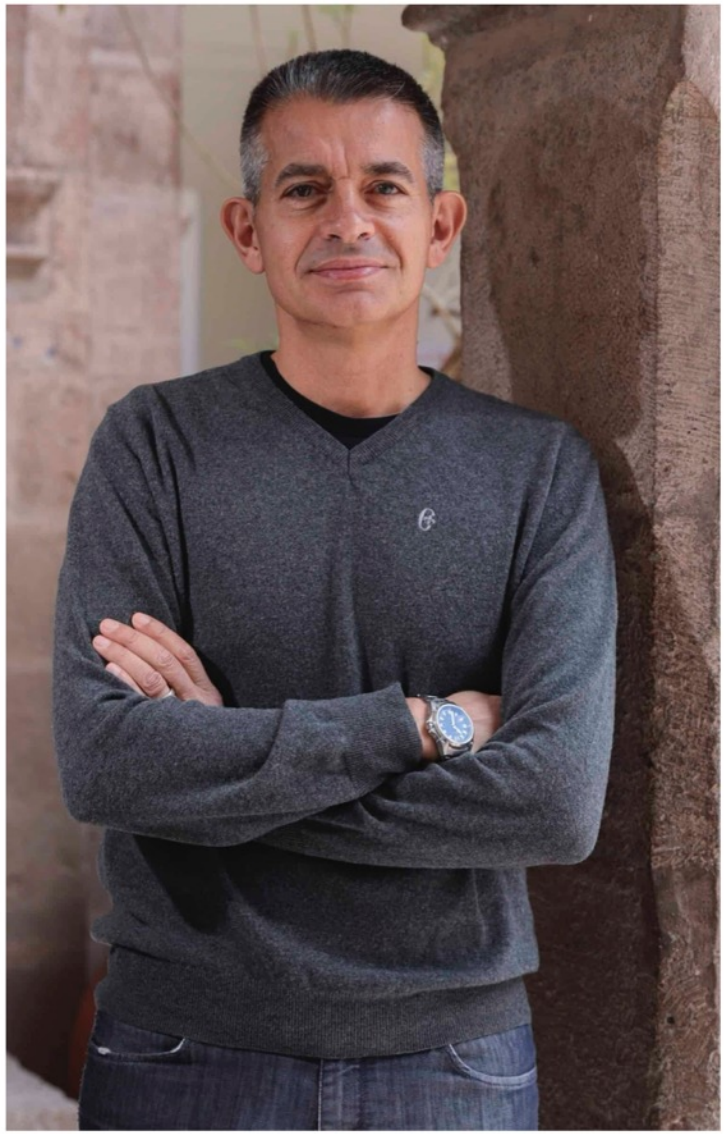
company/pulp-platform

youtube.com/pulp_platform

PULP Platform

Open Source Hardware, the way it should be!

PULP team at ETH Zürich: Open-source HW since 2013



- **Led by Luca Benini**
  - Professor at ETH Zürich and University of Bologna
- **Large team of around 100 people**

- **Parallel Ultra Low Power (PULP) platform**
  - https://pulp-platform.org/
  - https://x.com/pulp_platform

ETH zürich

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

We have designed over 60 ASICs using open-source HW

All our designs are based on open-source HW published on our GitHub page

- All using a permissive open source license (SolderPad)

https://github.com/pulp-platform

See our chip gallery under: http://asic.ethz.ch/

End-to-end Open-Source IC Design is possible today!

Design: from PULP

`github.com/pulp-platform`

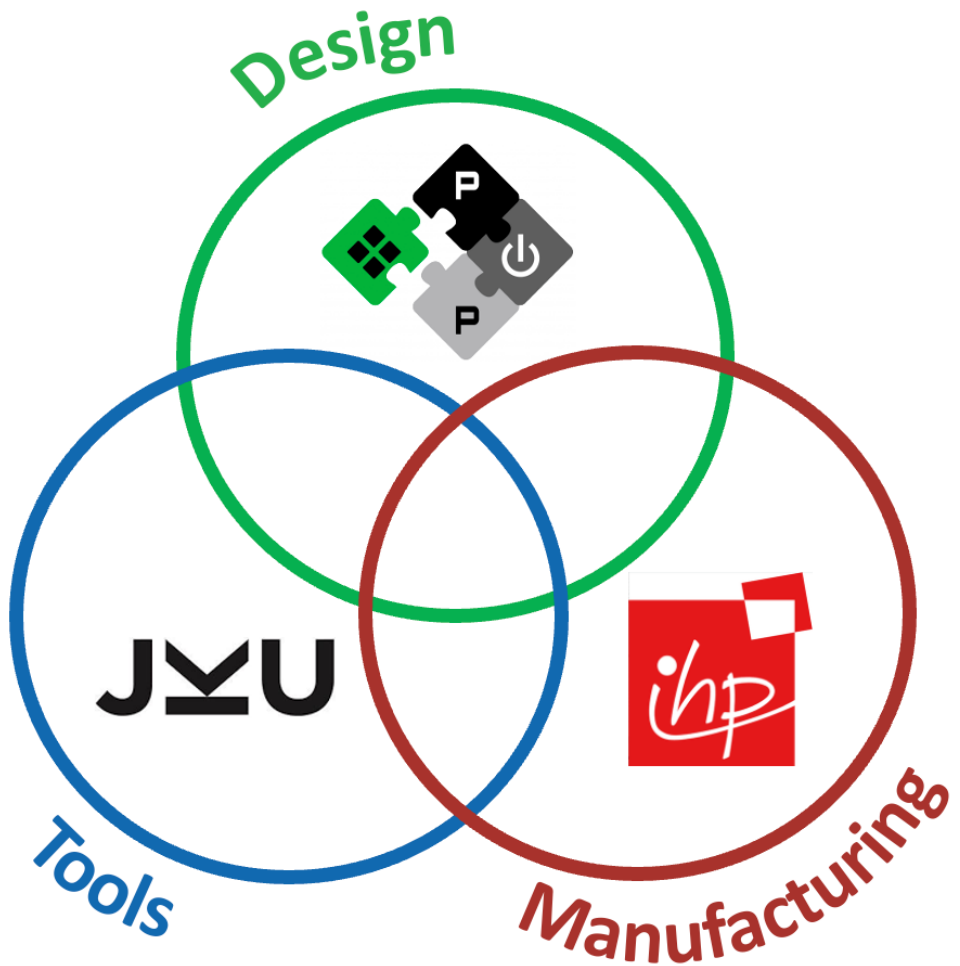
Tools: from Johannes Kepler University (JKU)

Reliable VM with large collection of open-source tools

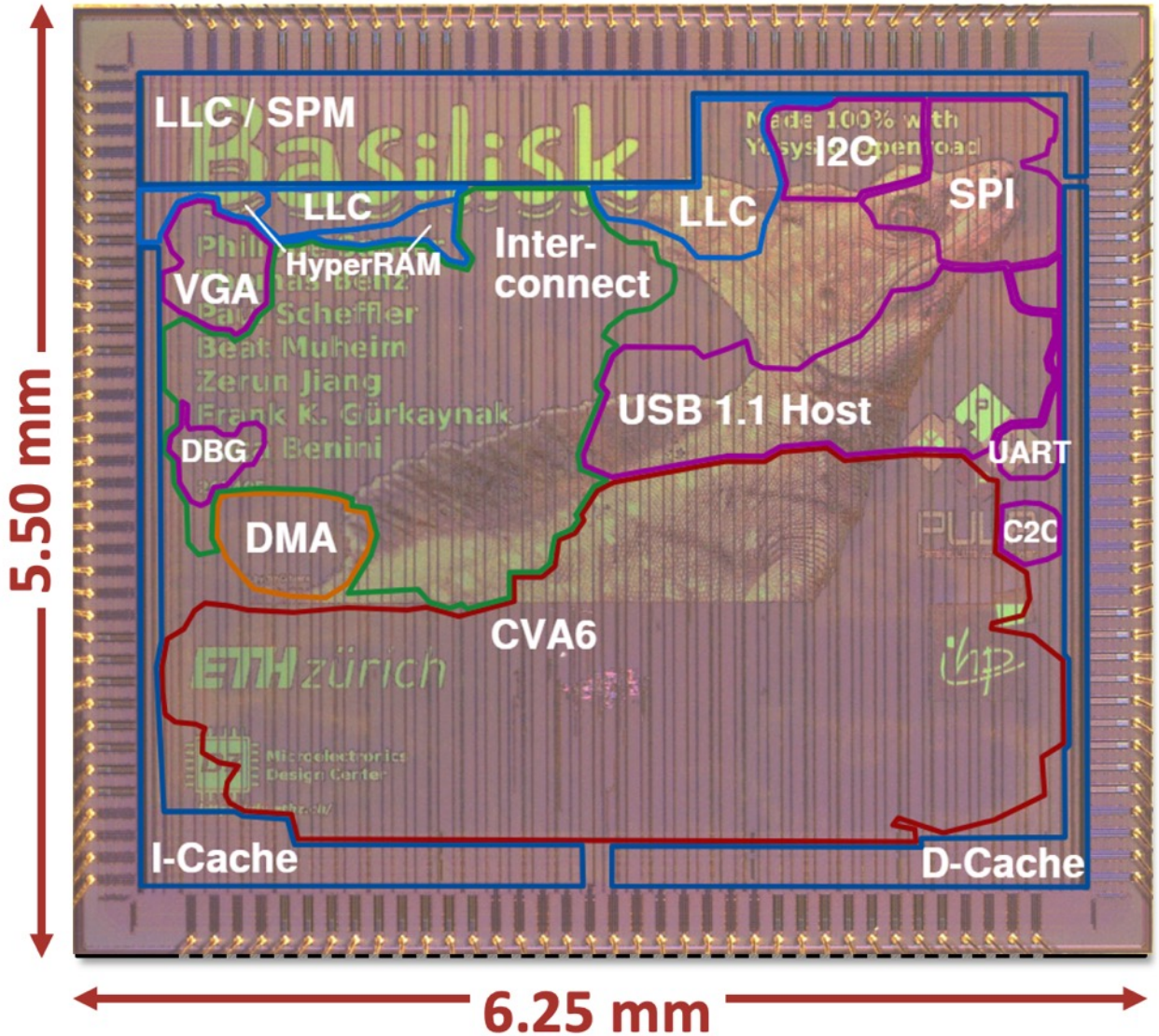
`github.com/iic-jku/IIC-OSIC-TOOLS`

Manufacturing: IHP130nm

`github.com/IHP-GmbH/IHP-Open-PDK`



Basilisk: Open RTL, Open EDA, Open PDK



- **Designed in IHP 130nm OpenPDK with Yosys, OpenROAD, KiCAD (board)**
  - SV-to-Verilog chain @ **<2min** runtime
  - Yosys synthesis:
    → **1.1 MGE** (**1.6×**) @ **77 MHz** (**2.3×**)
    → **1.4×** less runtime, **2.4×** less peak RAM
  - OpenROAD P&R tuning:
    → **-12%** die area, **+10%** core utilization

PULP

github.com/pulp-platform/cheshire-ihp130-o