ETH zürich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Challenges with Basilisk and where the Future is going

Integrated Systems Laboratory (ETH Zürich)

Philippe Sauter	**phsauter@iis.ee.ethz.ch**
Thomas Benz	tbenz@iis.ee.ethz.ch
Paul Scheffler	paulsc@iis.ee.ethz.ch
Frank K. Gürkaynak	kgf@iis.ee.ethz.ch
Luca Benini	lbenini@iis.ee.ethz.ch

PULP Platform

Open Source Hardware, the way it should be!

@pulp_platform

pulp-platform.org

youtube.com/pulp_platform

The PULP Open-Source Ecosystem

RISC-V Cores and Vector Units

RI5CY *CV32E*	Zero R *Ibex*	Snitch	Spatz	Ariane *CVA6*	ARA
RV32	RV32	RV32	RVV	RV64	RVV

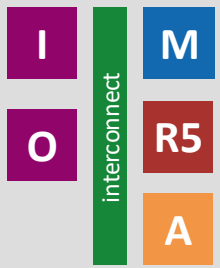
Peripherals

JTAG, SPI, UART, I2S, DMA, GPIO

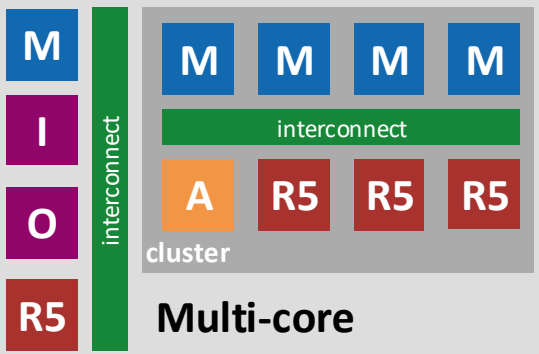
Interconnects

LIC, HCI, APB, FlooNoC, AXI4

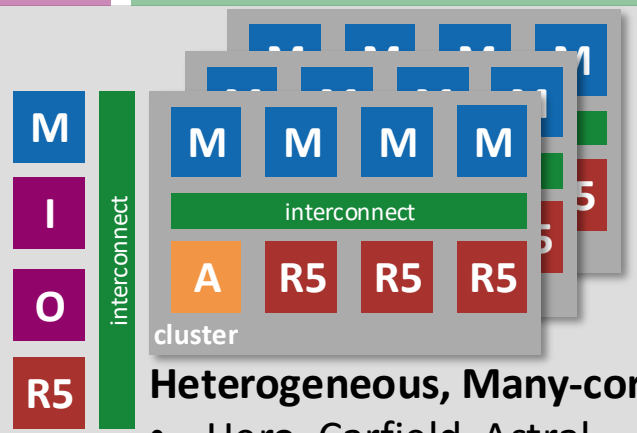
Platforms



Single core
- PULPino, PULPissimo
- **Cheshire**



Multi-core
- OpenPULP
- ControlPULP



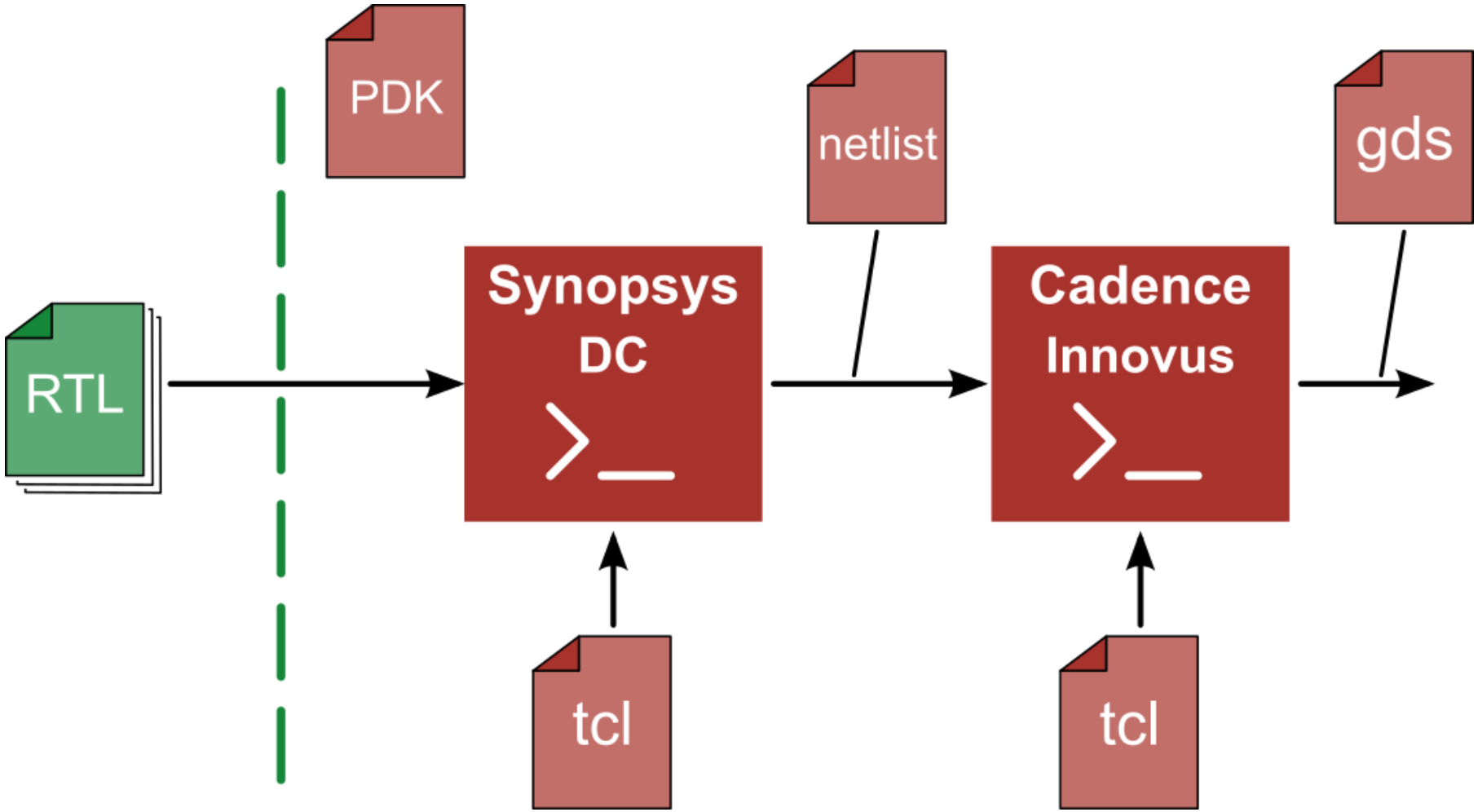
Heterogeneous, Many-core
- Hero, Carfield, Astral
- Occamy, Mempool

IOT → **HPC**

Accelerators and ISA extensions

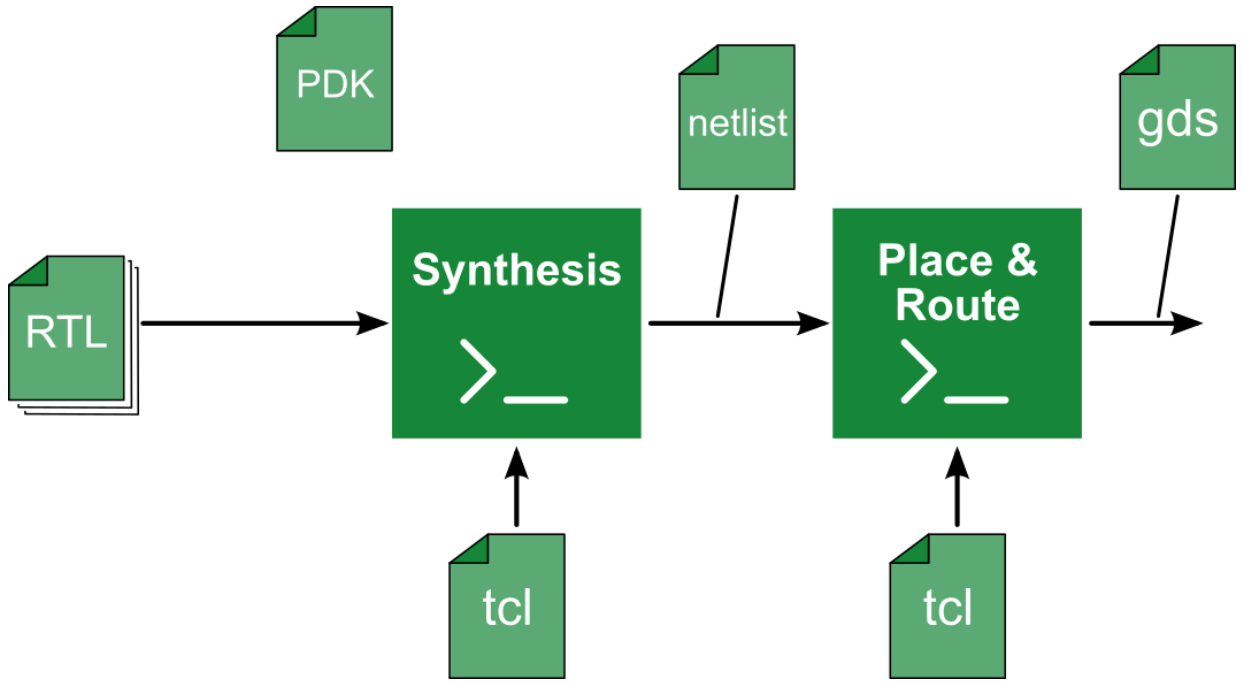
XpulpNN, XpulpTNN	ITA (Transformers)	RBE, NEUREKA (QNNs)	FFT (DSP)	REDMULE (FP-Tensor)

The Year is 2023, PULP is Open-Source



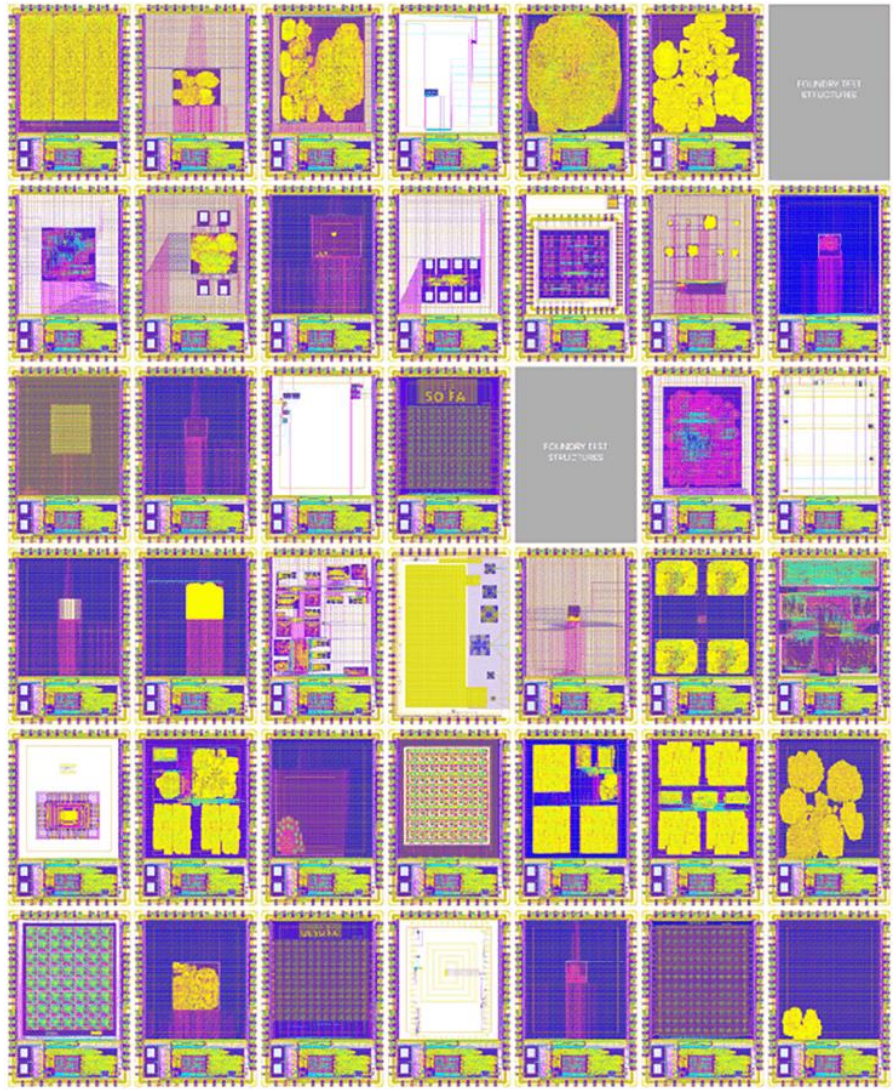
Why are we Exploring more Openness?

- **Research**
  - Easier collaboration
  - Reproducing results
  - New research (using tools or data)
- **Education**
  - Increased access
  - Experiment with flows and tools
  - A look "under the hood"
- **Industry**
  - Transparent Chain-of-Trust
  - Lower Cost



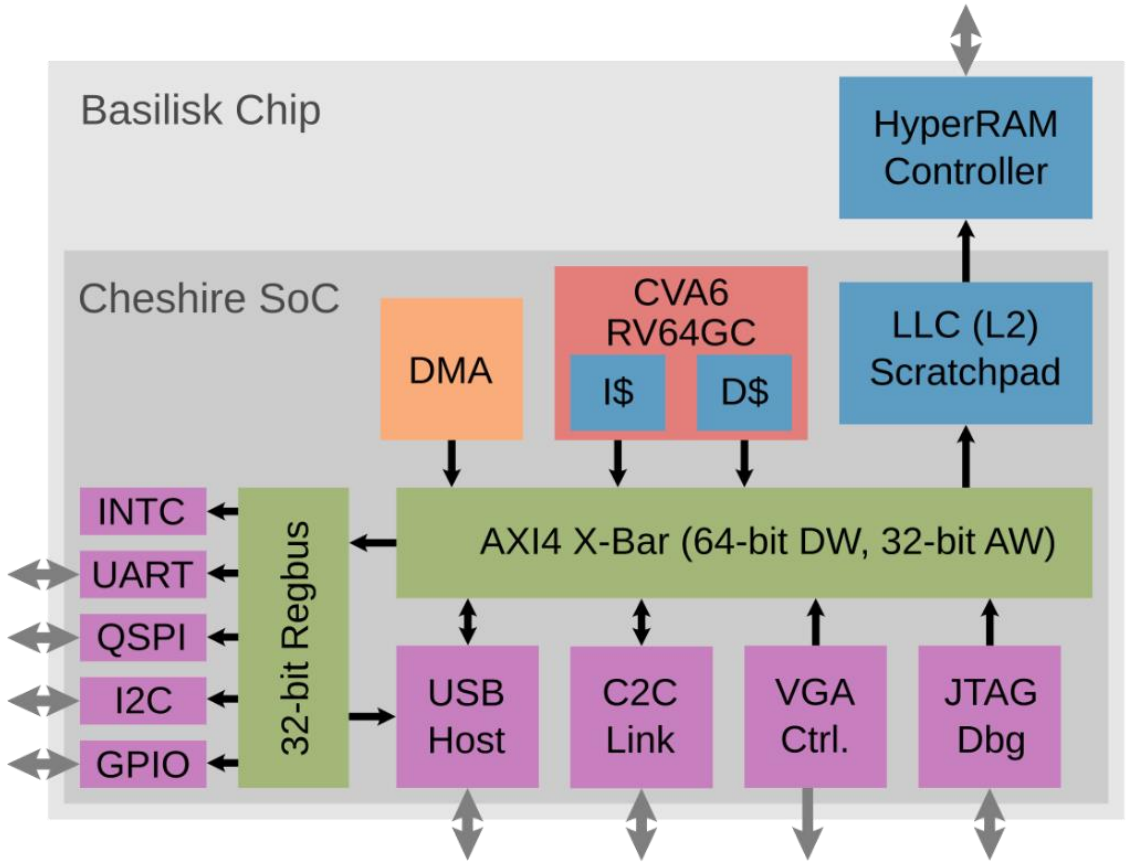
What Have Others Done?

- **Over 600 tapeouts**
  - Caravel Multi Project Wafers
  - Most <50kGE, <10% utilization
  - Big ones 200kGE
- **10x smaller than Cheshire**
- **Claim: Works “out of the box”**
  - True for existing designs but…
- **We want to go further!**
  - **Build on what is already there**



Find the Limits with our Cheshire SoC Platform

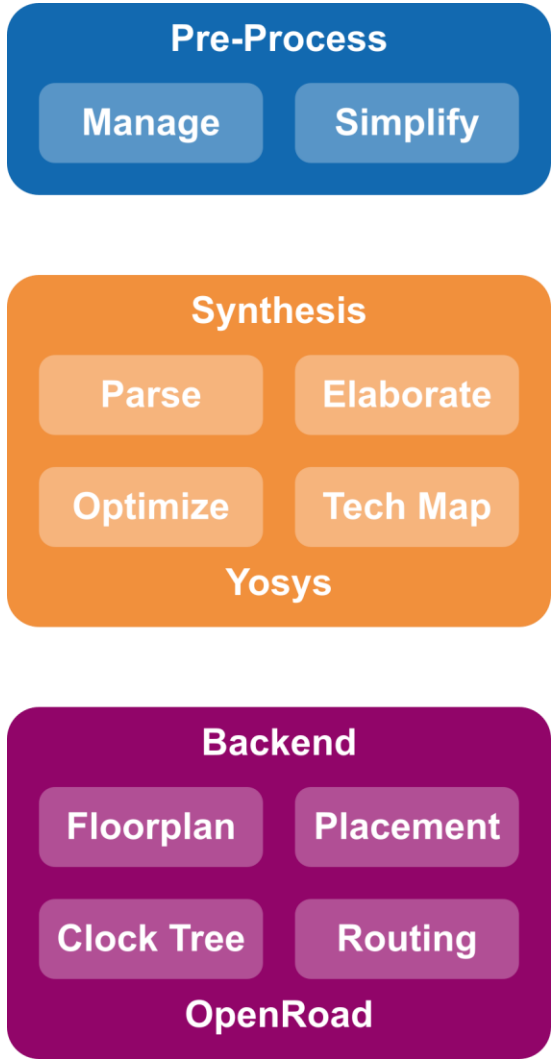
- **Multi-million gate design**
- **64-bit RISC-V Core**
  - Complete Linux-capable SoC
  - Simple “Raspberry Pi”
- **Rich Peripherals**
  - Including an open USB 1.1 host
- **Open-source DRAM interface**
  - Double data rate (DDR)
- **Silicon-proven**
  - Multiple successful tapeouts with closed-source commercial EDA



Ideal benchmark design to stress OS EDA

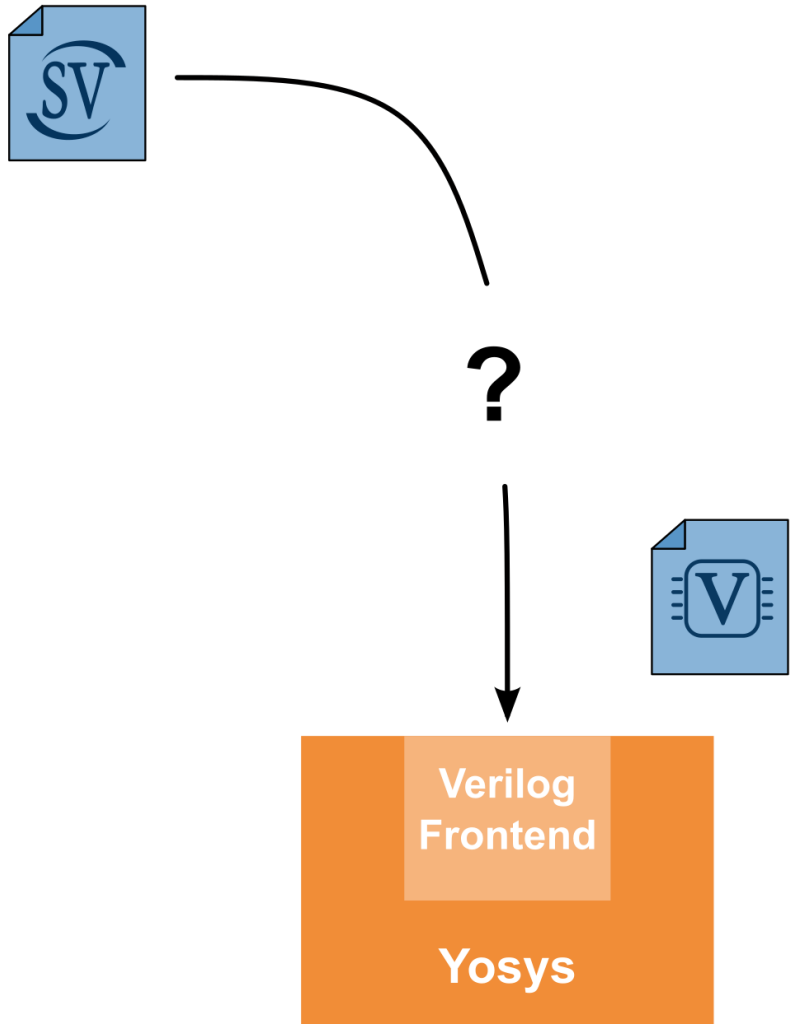
The Basic Battle Plan

- Get **PULP IPs into the synthesis tool**
  - If needed, simplify SystemVerilog
- **Synthesize the design**
  - Yosys takes it from RTL to generic-cells
  - Calls ABC for logic optimization and mapping to the technology (IHP 130nm)
- **Implement the designs backend**
  - Collection of research tools into OpenRoad
  - Places the cells and routes wires between them
- *Lets look at a few challenges we encountered*



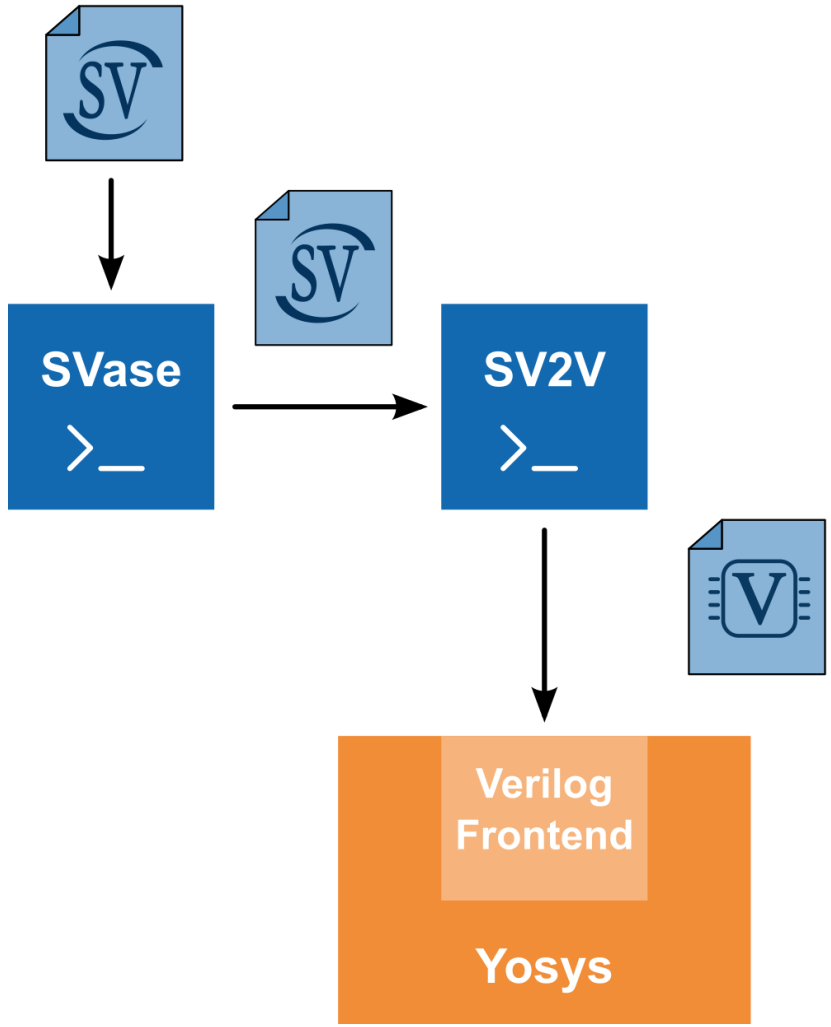
First Challenge: How do we Read the Code?

- **SystemVerilog is widespread**
  - **PULP-Platform: Cheshire**
  - lowRISC: OpenTitan and its IPs
  - OpenHW Group: CVA6, CV32E40P
  - BlackParrot RISC-V core
  - Most **industrial IPs** are implemented in SV



First Challenge: How do we Read the Code?

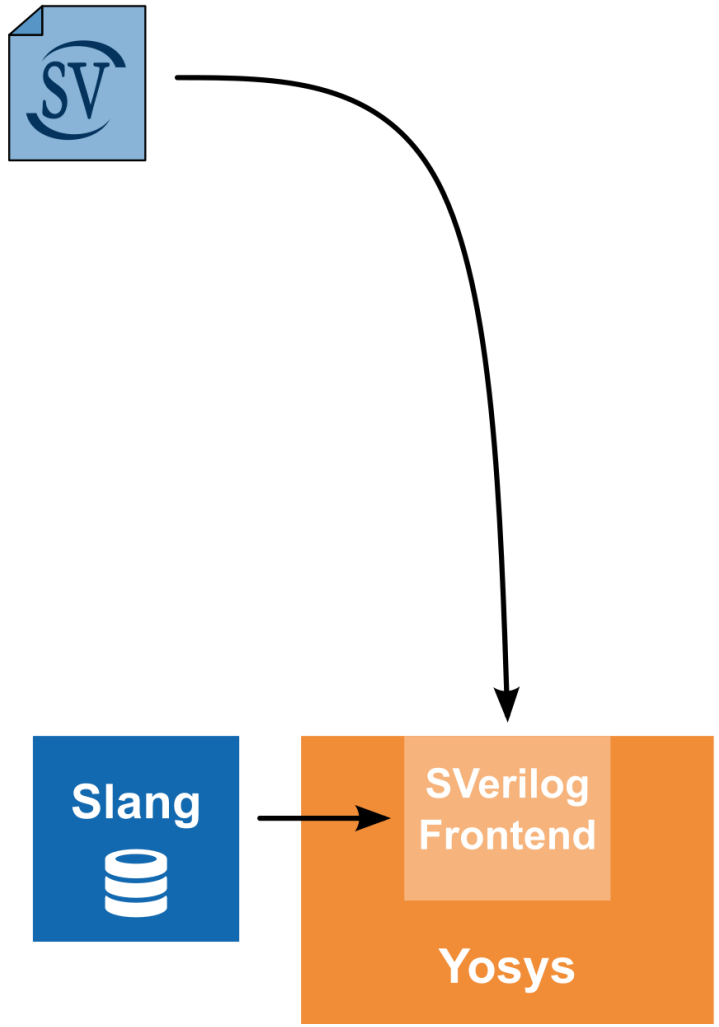
- **SystemVerilog is widespread**
  - **PULP-Platform: Cheshire**
  - lowRISC: OpenTitan and its IPs
  - OpenHW Group: CVA6, CV32E40P
  - BlackParrot RISC-V core
  - Most **industrial IPs** are implemented in SV
- **For Basilisk:**
  - Pre-process the RTL
  - SVase simplifies SystemVerilog
  - SV2V converts to Verilog



First Challenge: How do we Read the Code?

- **SystemVerilog is widespread**
  - **PULP-Platform: Cheshire**
  - lowRISC: OpenTitan and its IPs
  - OpenHW Group: CVA6, CV32E40P
  - BlackParrot RISC-V core
  - Most industrial IPs are implemented in SV
- For Basilisk:
  - Pre-process the RTL
  - Svase simplifies SystemVerilog
  - SV2V converts to Verilog

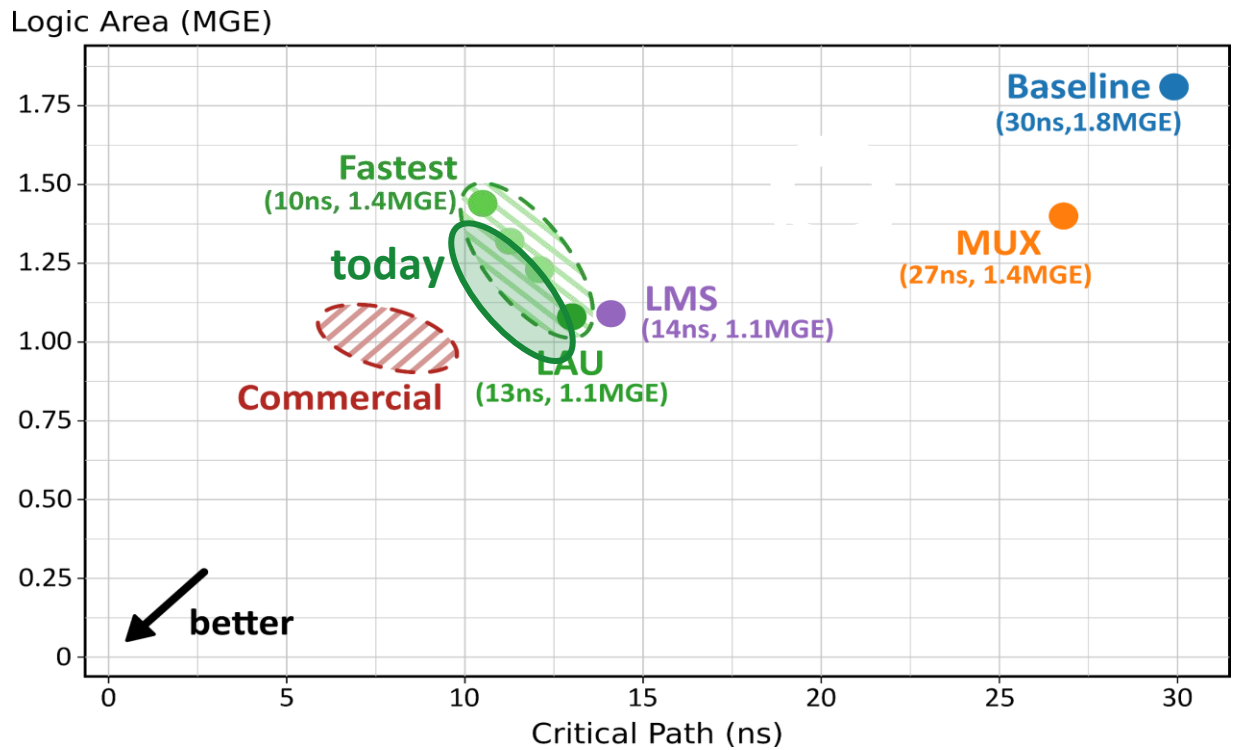
Reading SystemVerilog with one command using yosys-slang



github.com/povik/yosys-slang

Next Challenge: We need better Synthesis Results

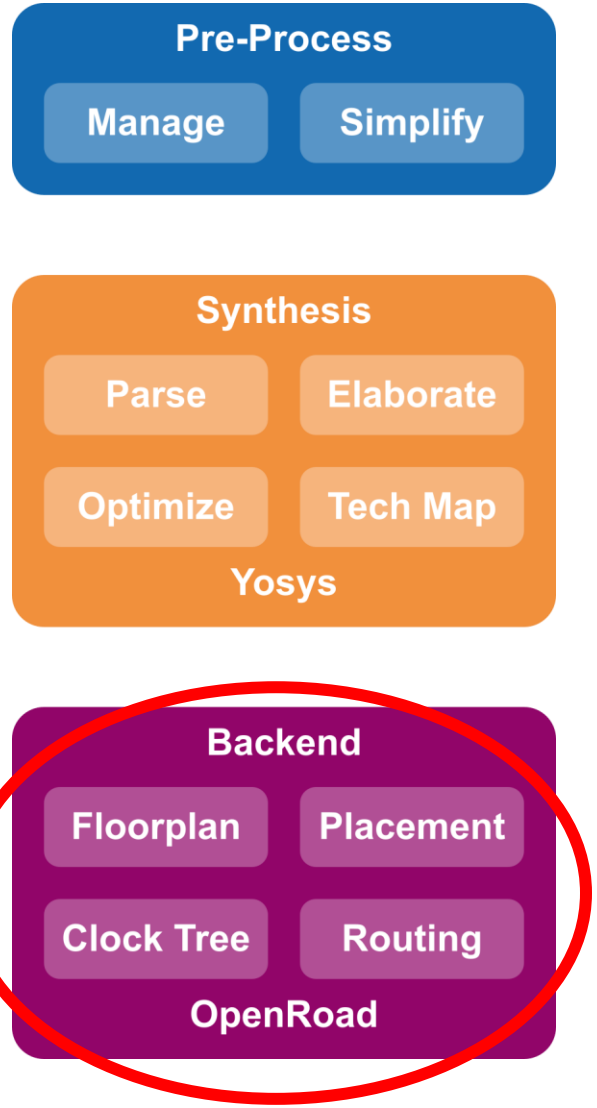
- **Improvements until June 2024**
  - SV-to-Verilog chain @ **<2min** runtime
  - Yosys synthesis:
    → **1.1 MGE** (**1.6×**) @ **77 MHz** (**2.3×**)
    → **1.4×** less runtime, **2.4×** less peak RAM
  - OpenROAD P&R: tuning
    → **-12%** die area, **+10%** core utilization
- **Improvements June-October**
  - Yosys-slang replaces SV2V
    - **1.6×** less runtime, **10×** less peak RAM
  - **-10%** logic area



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

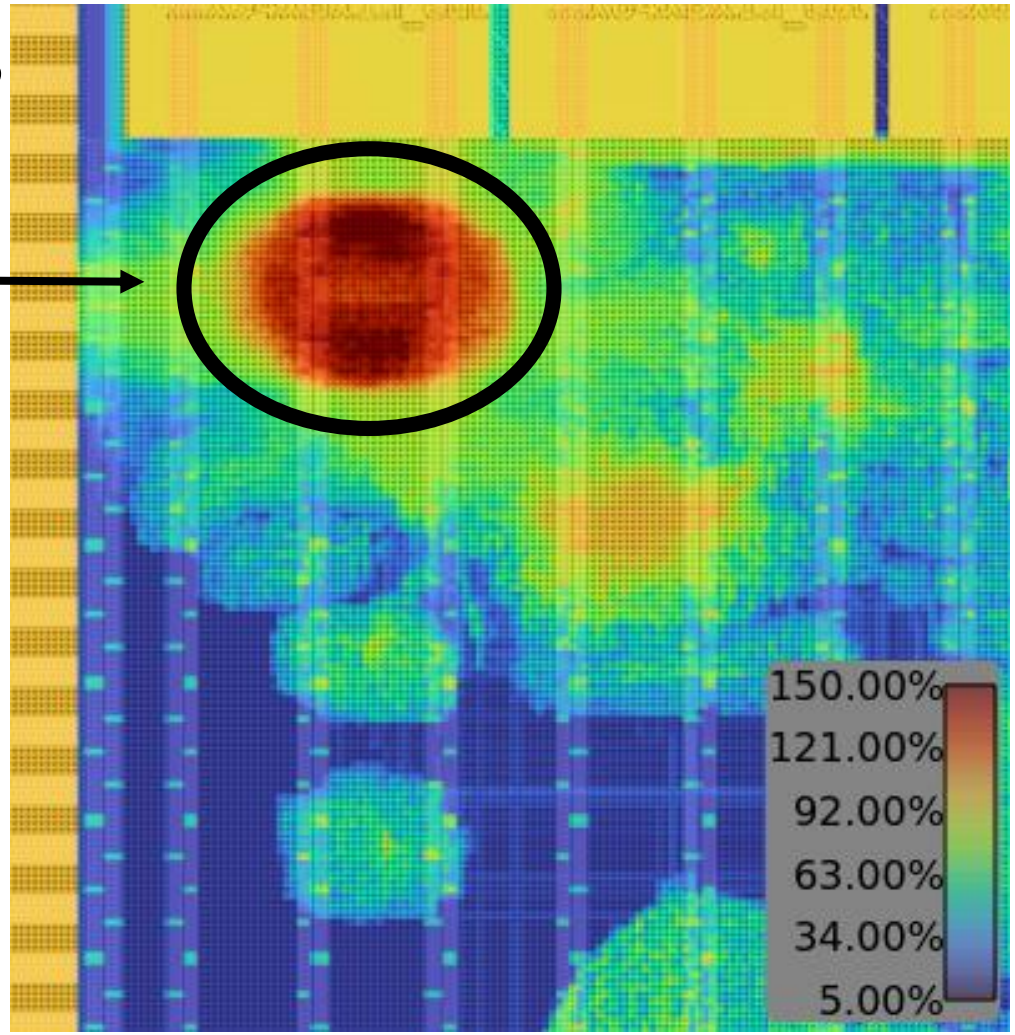
The Basic Battle Plan

- ~~Get **PULP IPs into the synthesis tool**~~
  - ~~If needed, simplify SystemVerilog~~
- ~~**Synthesize the design**~~
  - ~~Yosys takes it from RTL to generic cells~~
  - ~~Calls ABC for logic optimization and mapping to the technology (IHP 130nm)~~
- **Implement the designs backend**
  - Collection of research tools into OpenRoad
  - Places the cells and routes wires between them
- *Lets look at a few challenges we encountered*



Challenge: Cell Placement

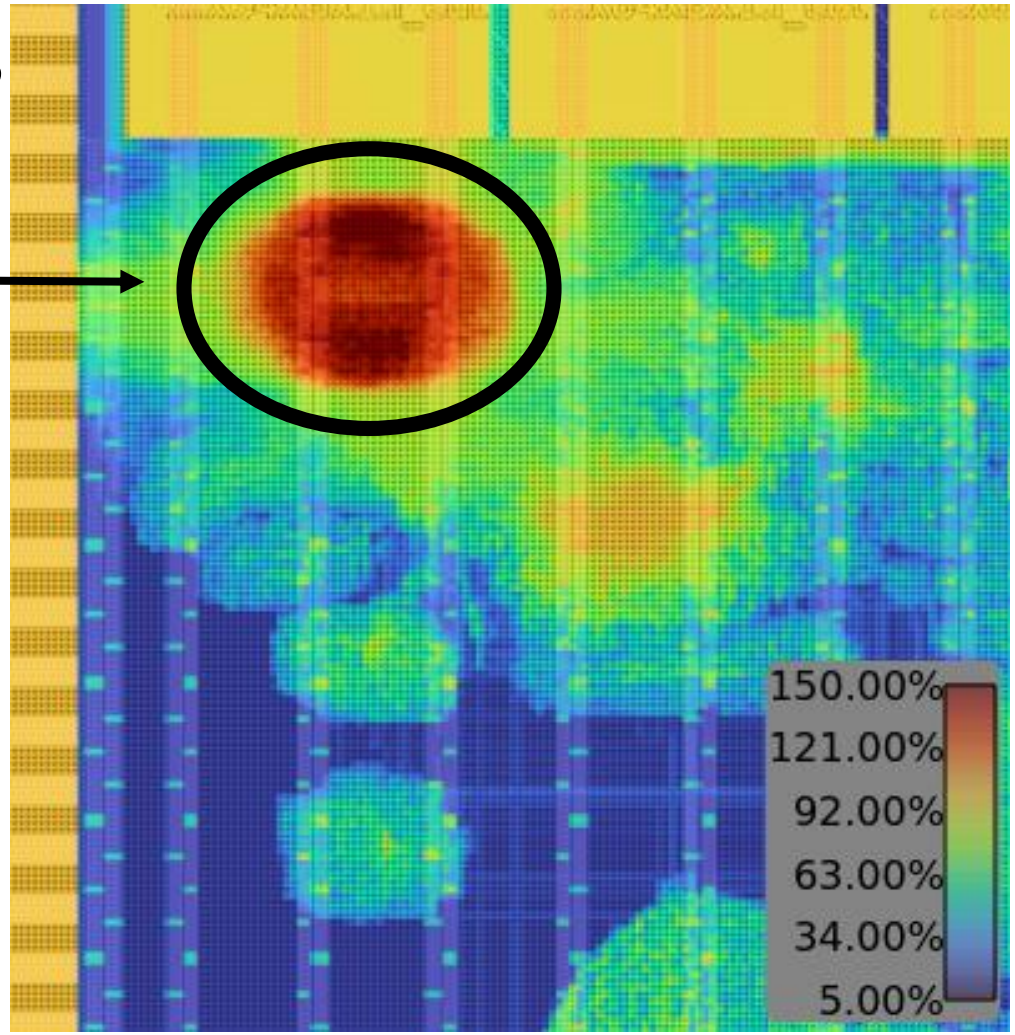
- ***Remember the Routing Congestion Task?***
- **High Congestion in Bootrom**



Challenge: Cell Placement

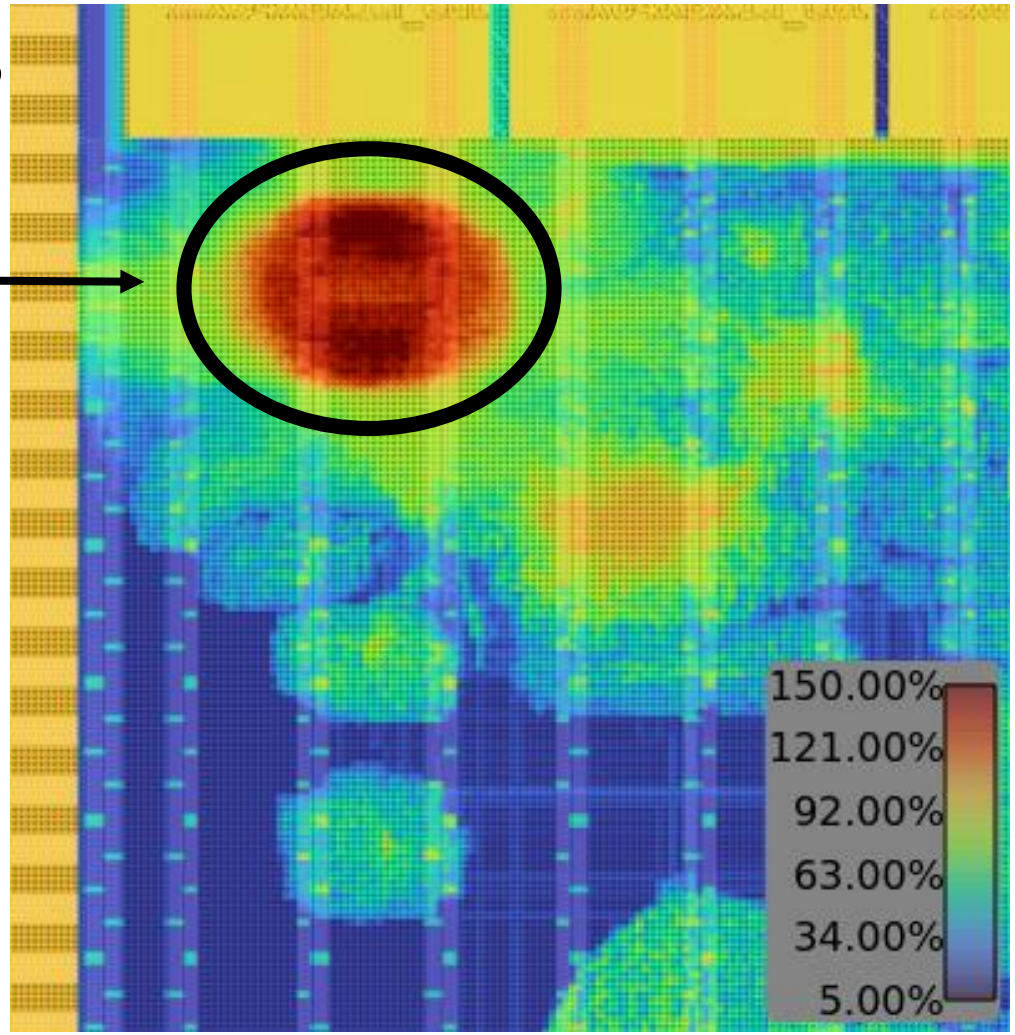
- ***Remember the Routing Congestion Task?***

- **High Congestion in Bootrom**
- **Decrease local density**
  - More routing resources per cell
  - Decreases congestion



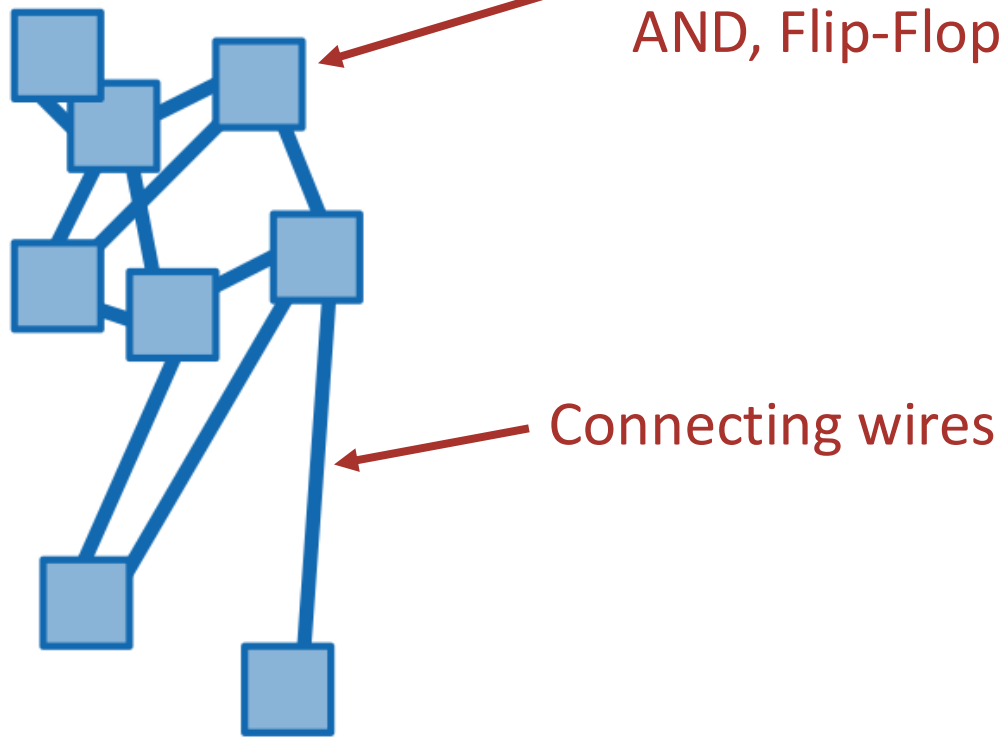
Challenge: Cell Placement

- ***Remember the Routing Congestion Task?***
- **High Congestion in Bootrom**
- **Decrease local density**
  - More routing resources per cell
  - Decreases congestion
- **OpenROAD**
  - Doesn't lower density enough
- ***Can we help OpenRoad?***



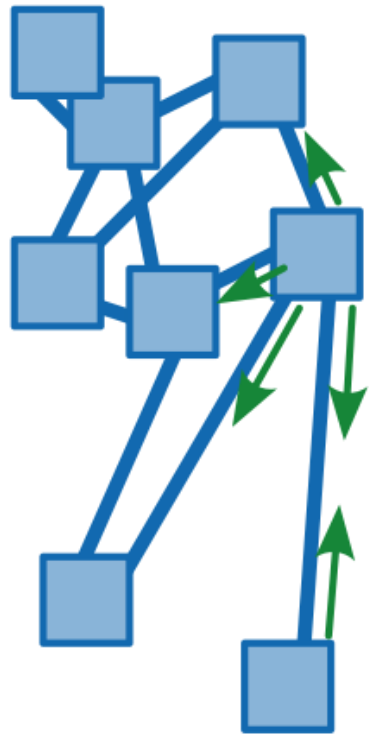
Attracting Wires, Repelling Cells

- **Goal:**
  - Minimize wire-length
  - Achieve target density



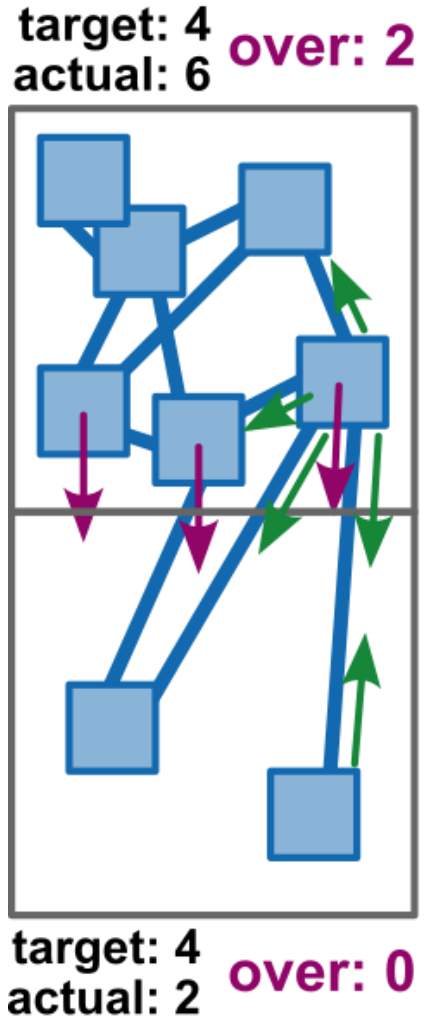
Attracting Wires, Repelling Cells

- **Goal:**
  - Minimize wire-length
  - Achieve target density
- **Wire-force attracts** connected cells



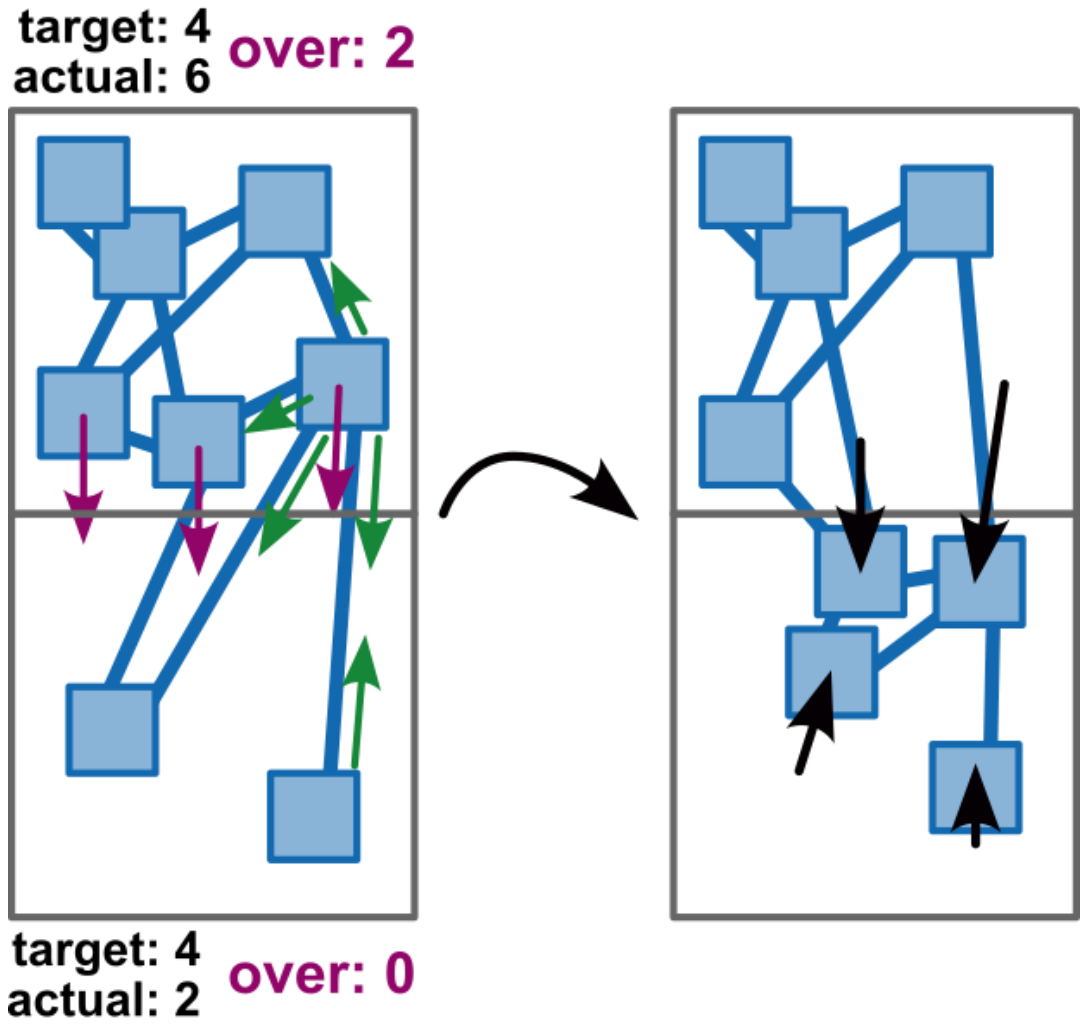
Attracting Wires, Repelling Cells

- **Goal:**
  - Minimize wire-length
  - Achieve target density
- **Wire-force attracts** connected cells
- **Density-force push** overflowing cells **away**



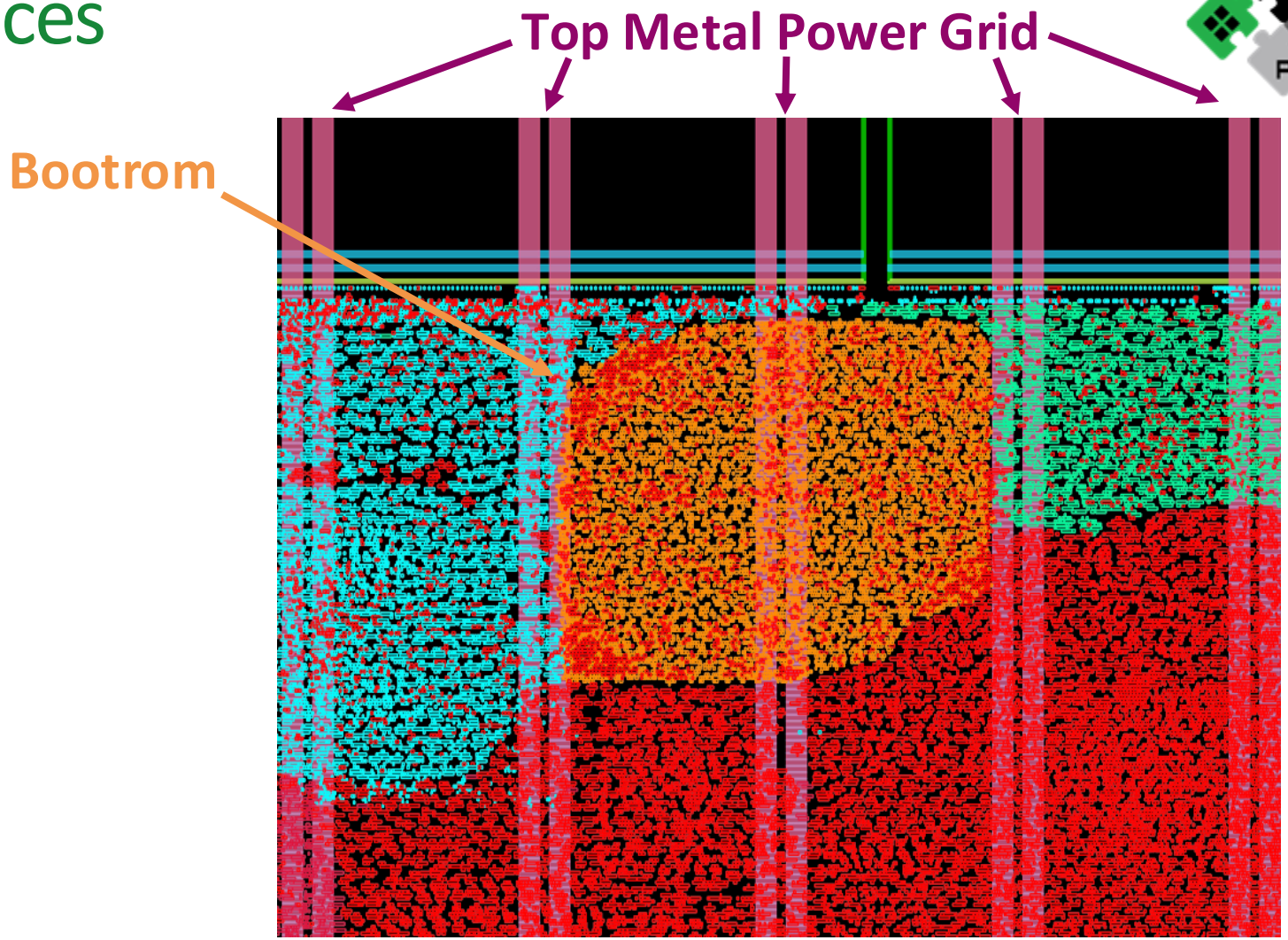
Attracting Wires, Repelling Cells

- **Goal:**
  - Minimize wire-length
  - Achieve target density
- **Wire-force attracts** connected cells
- **Density-force push** overflowing cells **away**
- **Total force per cell**
  - Cell starts moving in this direction
  - Incrementally adds velocity
- **Placement Tool**
  - Find useful force-balance
  - Weight critical nets/cells



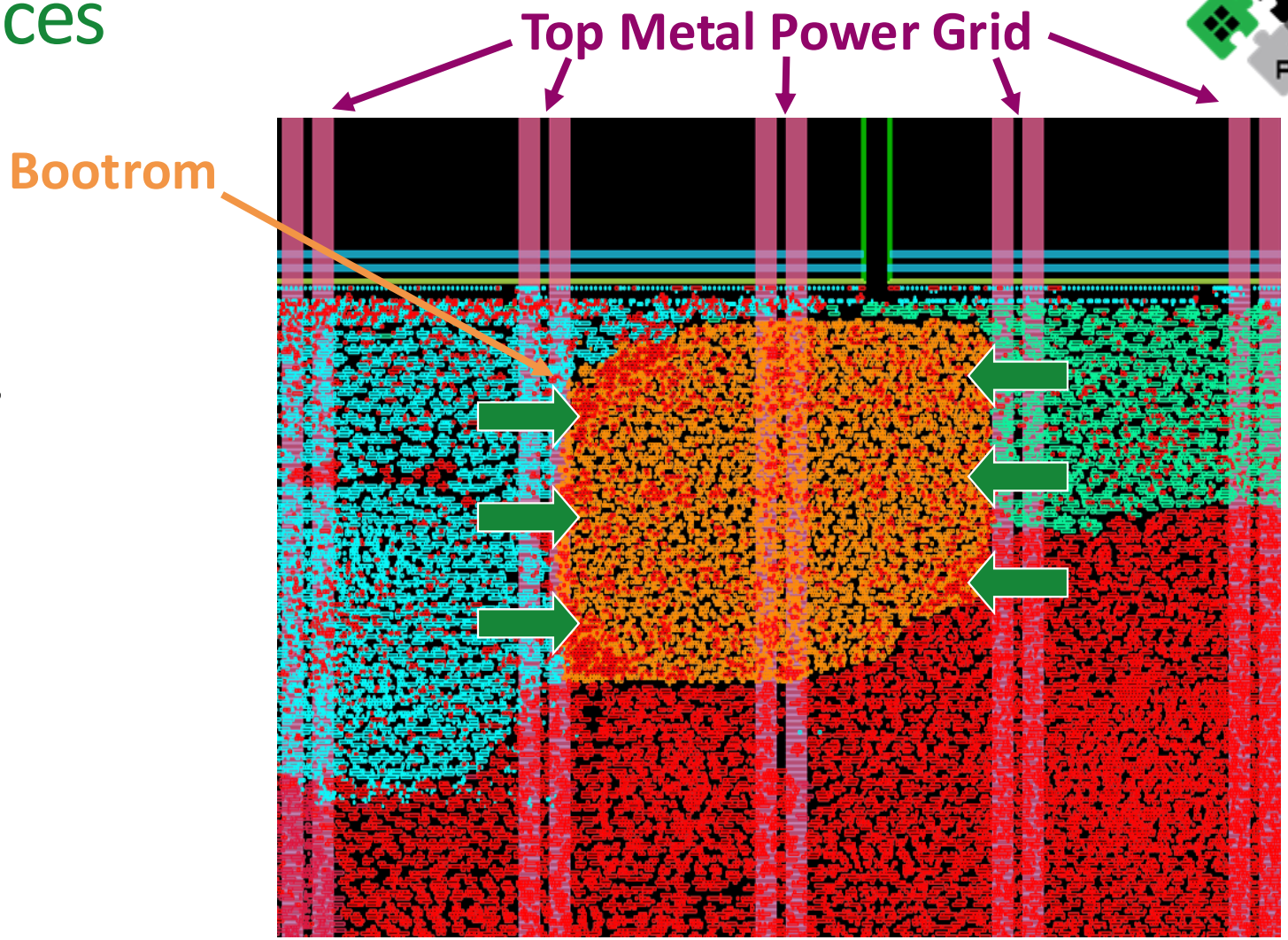
Power Stripes are Fences

- **Initial power stripes**
  - Wide VSS/VDD in pairs



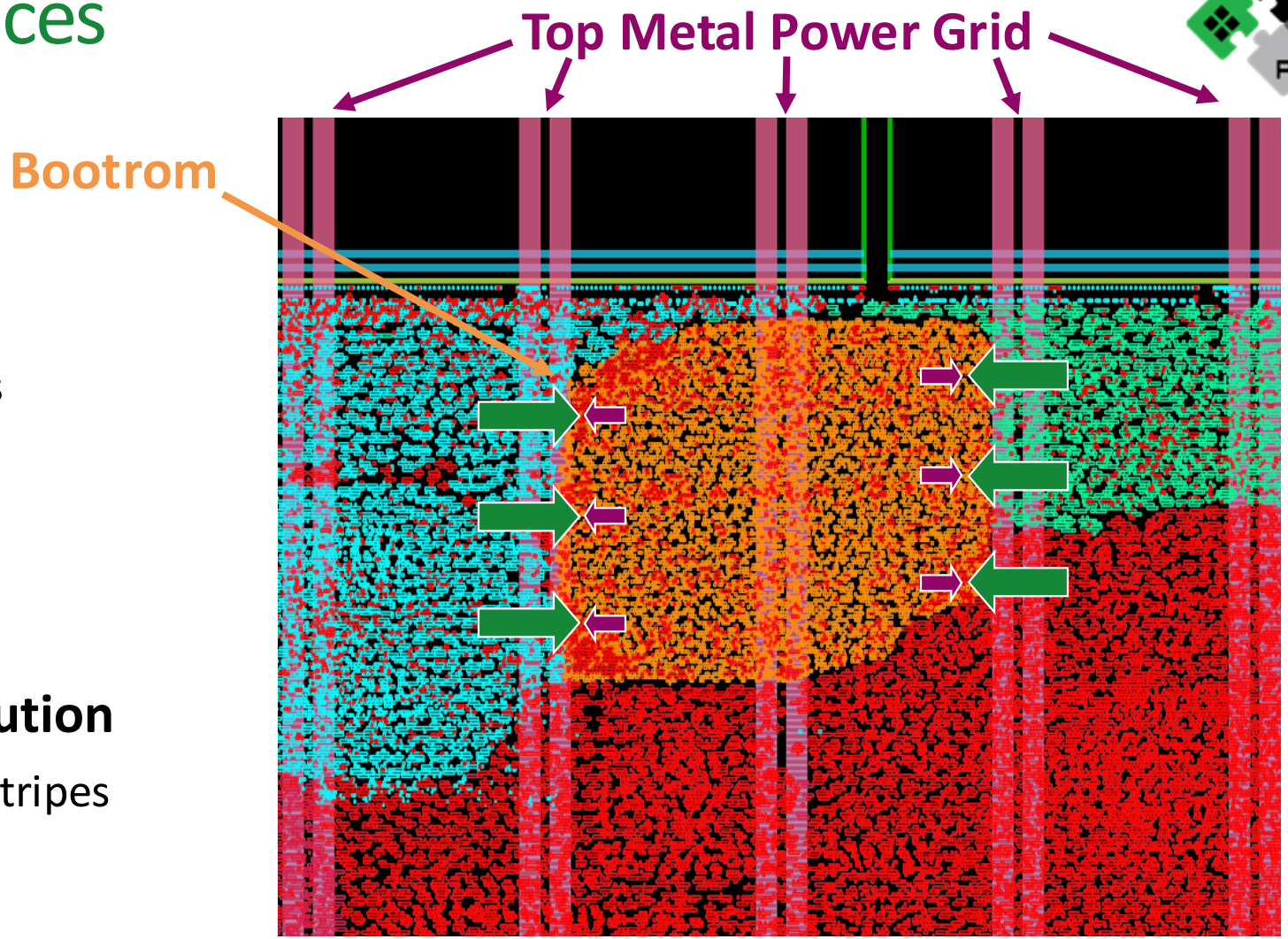
Power Stripes are Fences

- **Initial power stripes**
  - Wide VSS/VDD in pairs
- **Wire-force pulls in**
  - High penalty for crossing stripes



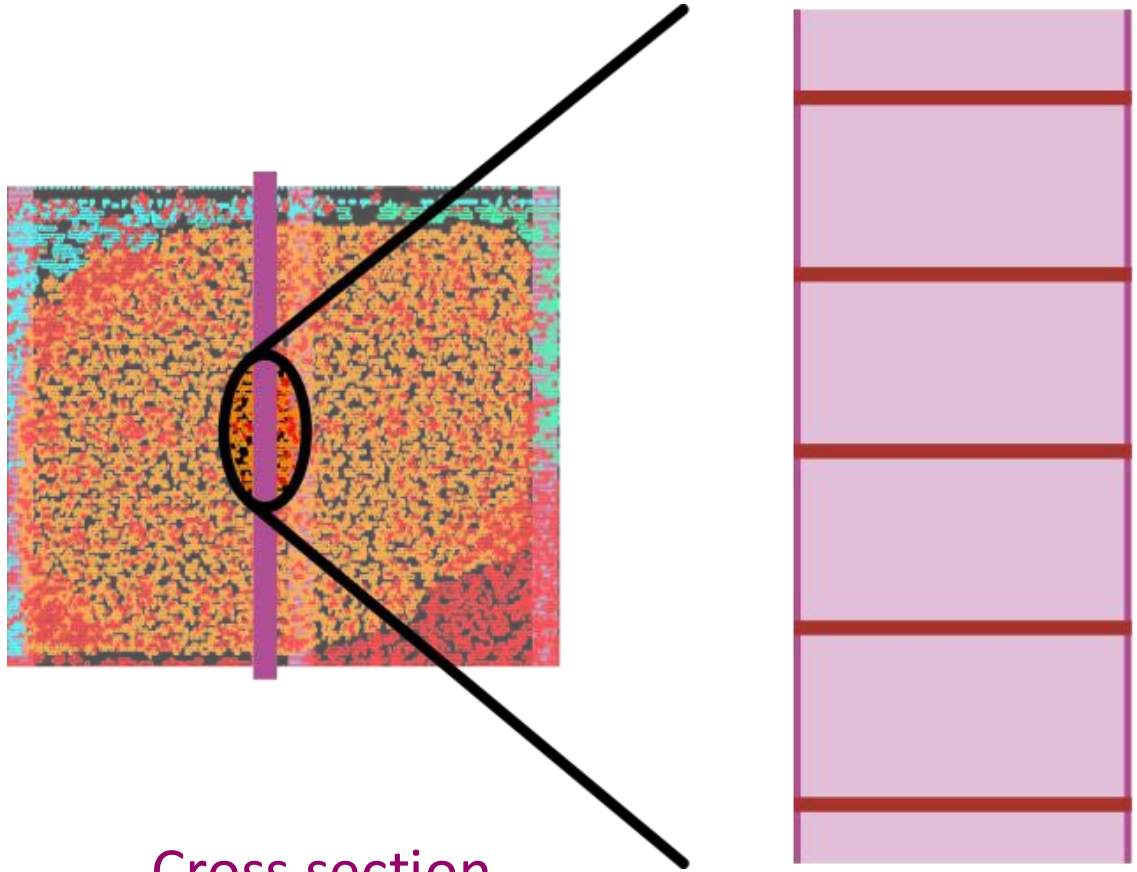
Power Stripes are Fences

- **Initial power stripes**
  - Wide VSS/VDD in pairs
- **Wire-force pulls in**
  - High penalty for crossing stripes
- **Density-force too small**
  - Trying to decrease local density
  - Pushes cells below stripes
- **Solution: more even distribution**
  - Thinner, more frequent power stripes



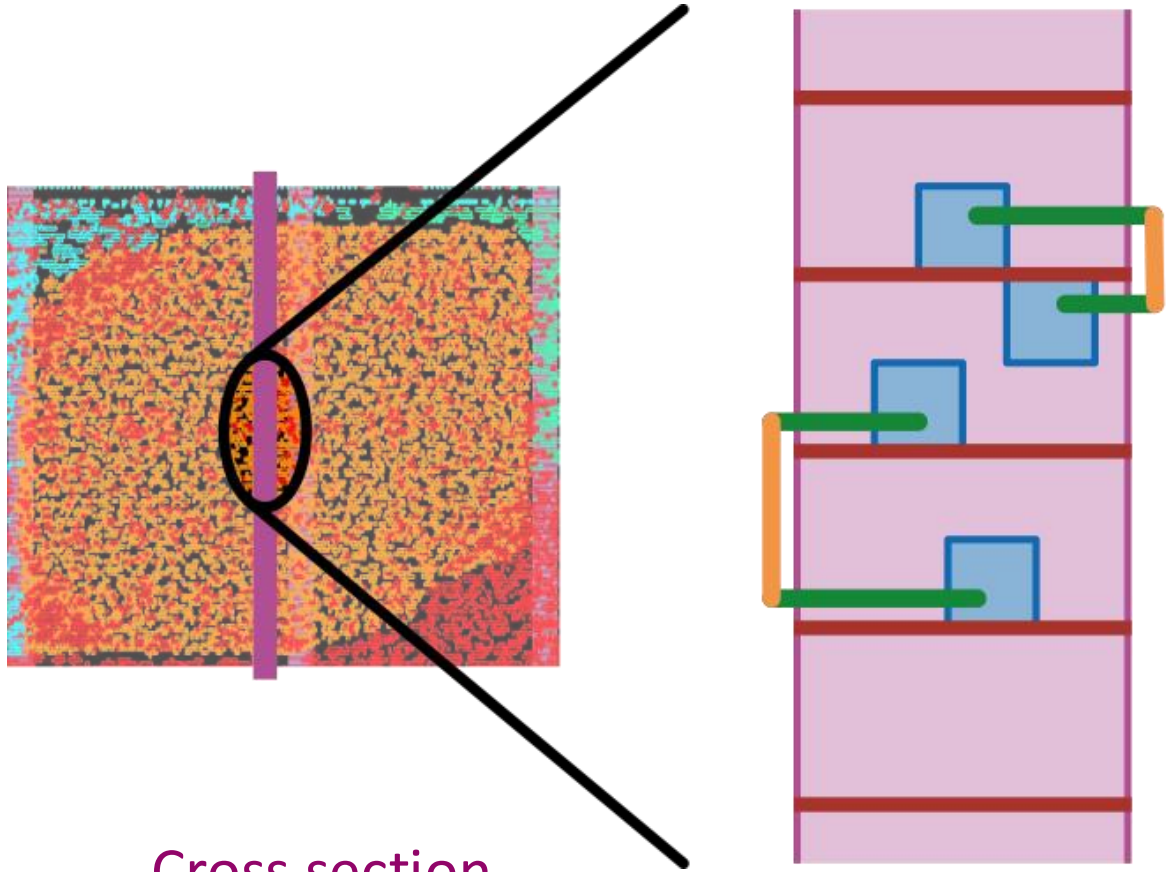
Power Stripes Block Wires

- **Power via-stack block routing**



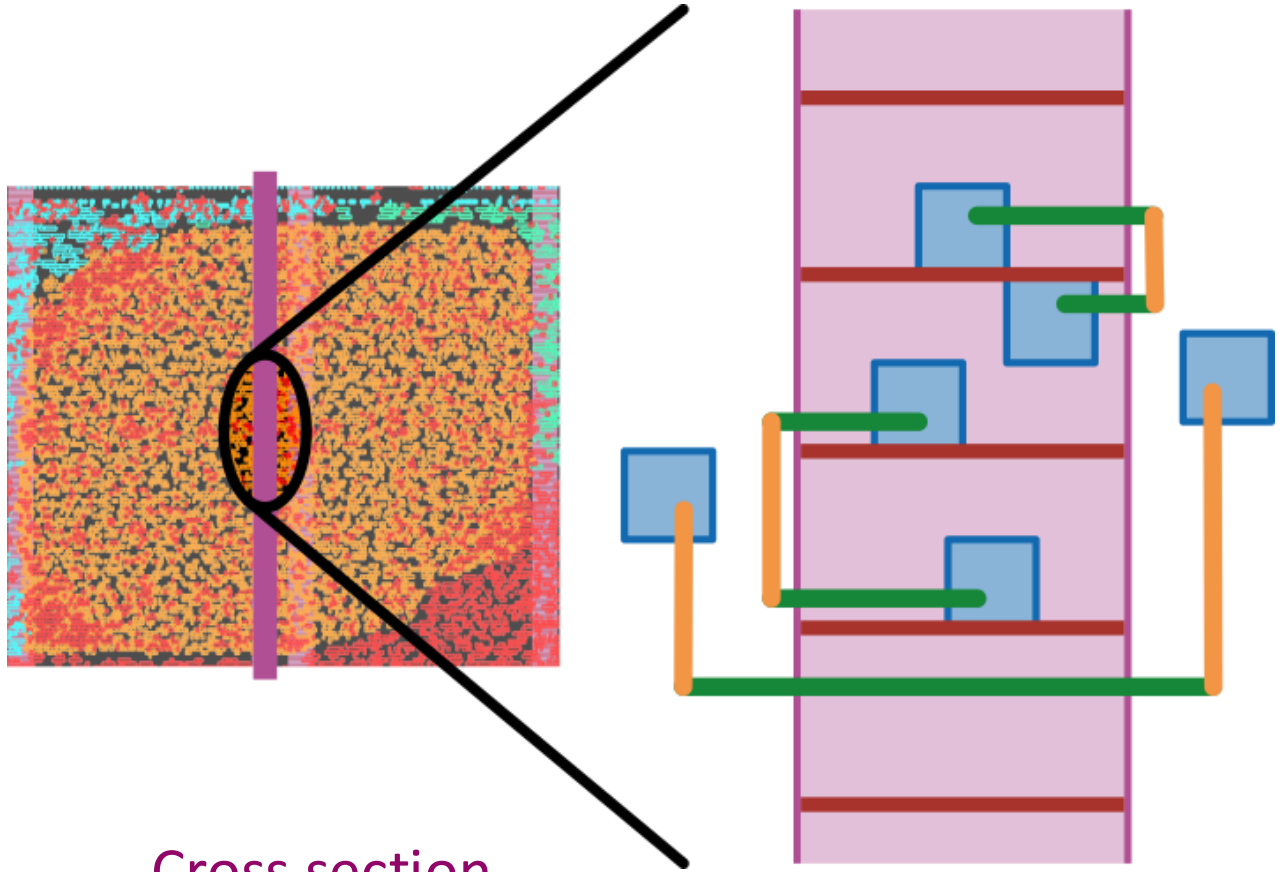
Power Stripes Block Wires

- **Power via-stack block routing**
- **Cells under stripe**
  - No vertical routing
  - Router needs to go around
  - Longer horizontal wires



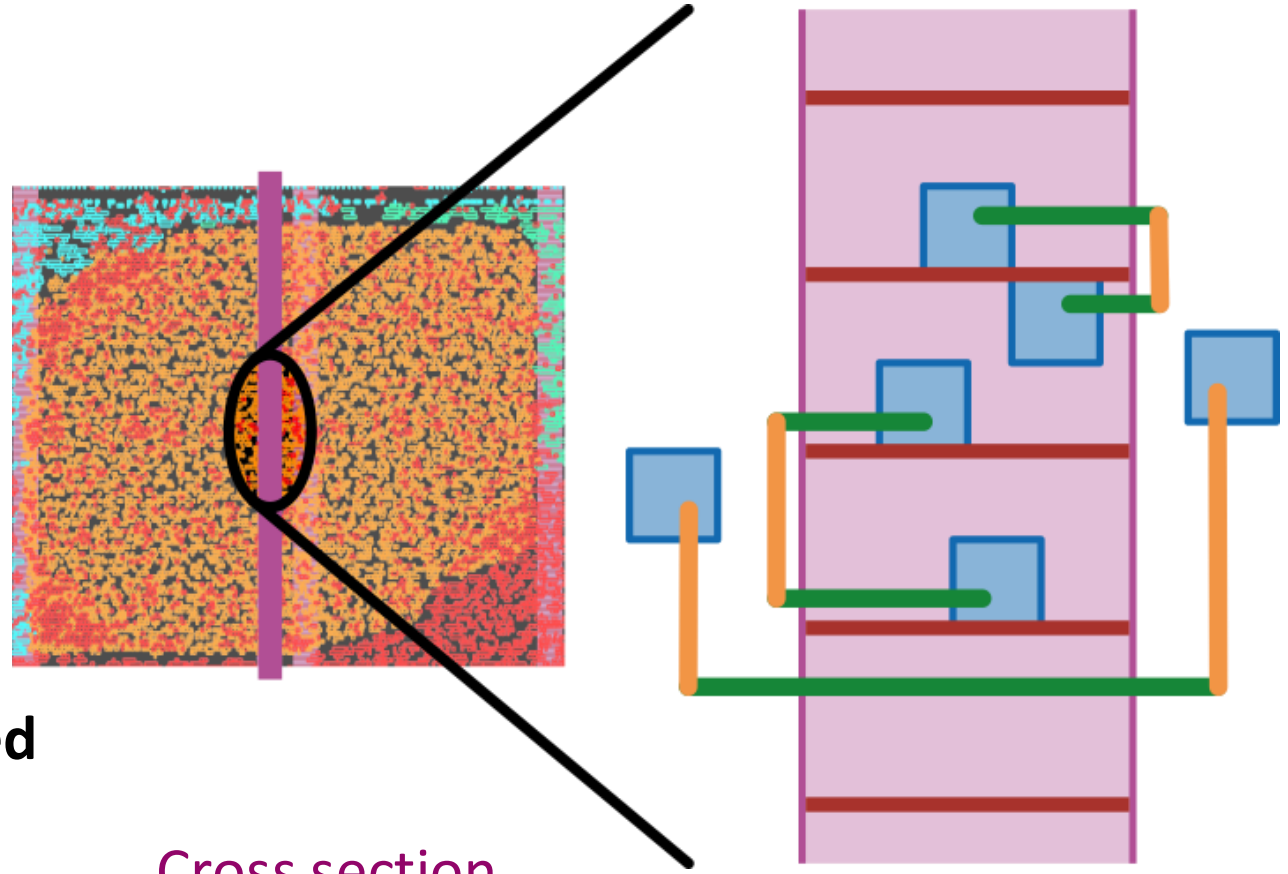
Power Stripes Block Wires

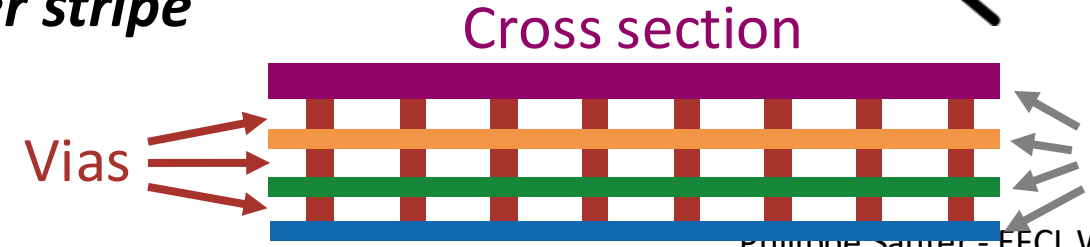
- **Power via-stack block routing**
- **Cells under stripe**
  - No vertical routing
  - Router needs to go around
  - Longer horizontal wires
- **Cells near stripe**
  - Goes to next unused opening
  - Longer vertical wires



Power Stripes Block Wires

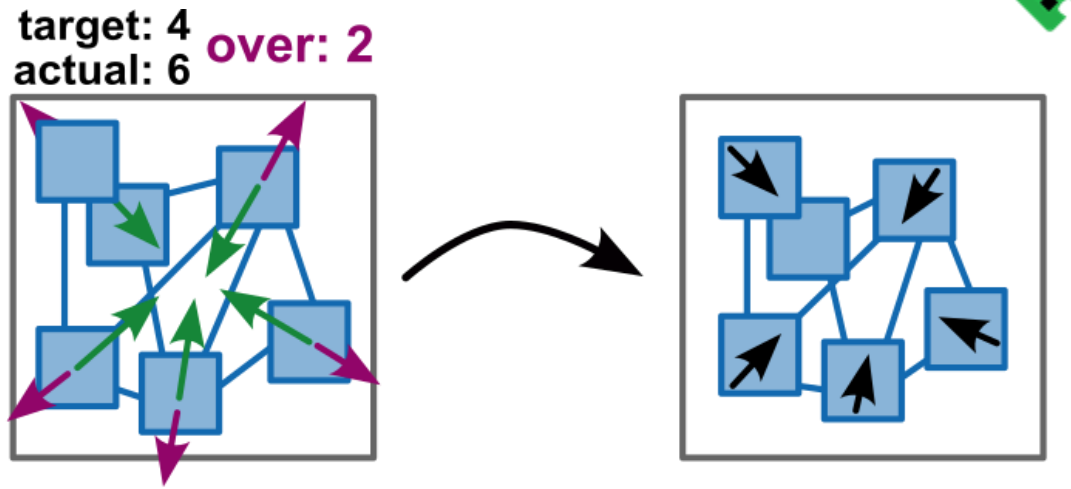
- **Power via-stack block routing**
- **Cells under stripe**
  - No vertical routing
  - Router needs to go around
  - Longer horizontal wires
- **Cells near stripe**
  - Goes to next unused opening
  - Longer vertical wires
- **Routing resources are overestimated**
- ***Solution: Blockage under stripe***





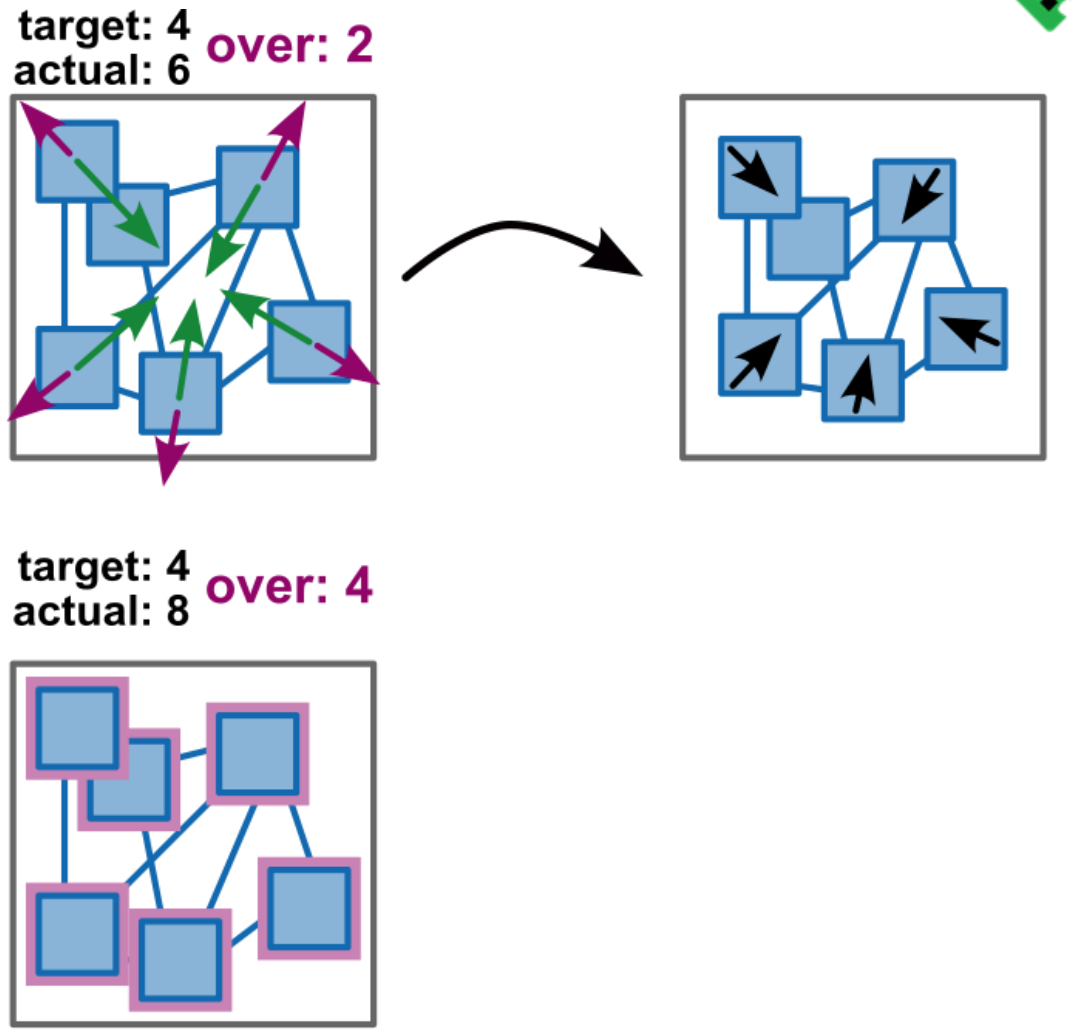
Parameter Tuning Required

- **Highly interconnected modules**
  - More wires per cell
  - Wire-force dominates
  - Pulls cells tightly together



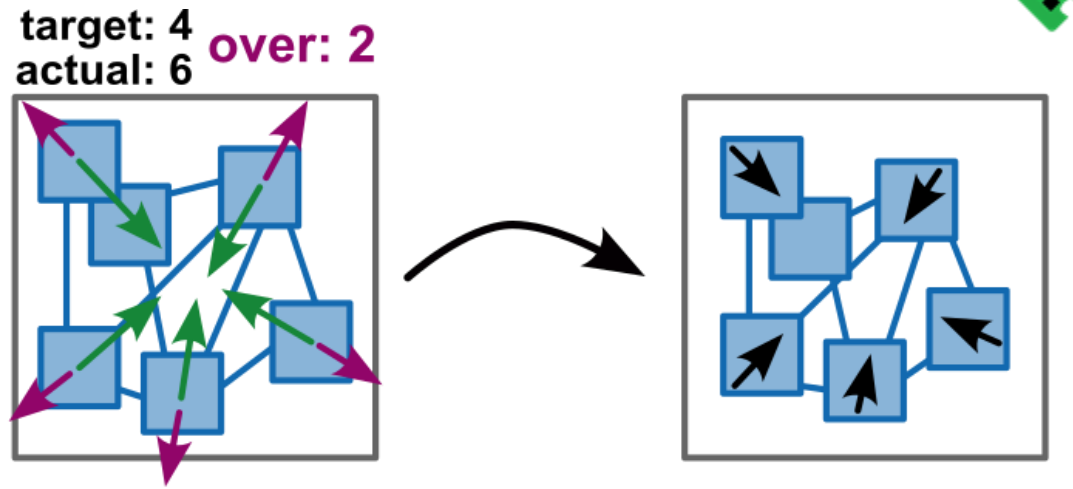
Parameter Tuning Required

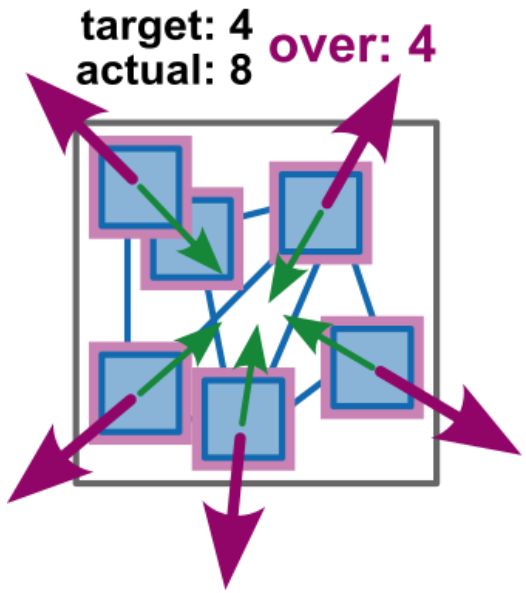
- **Highly interconnected modules**
  - More wires per cell
  - Wire-force dominates
  - Pulls cells tightly together
- **Solution: Routability driven**



Parameter Tuning Required

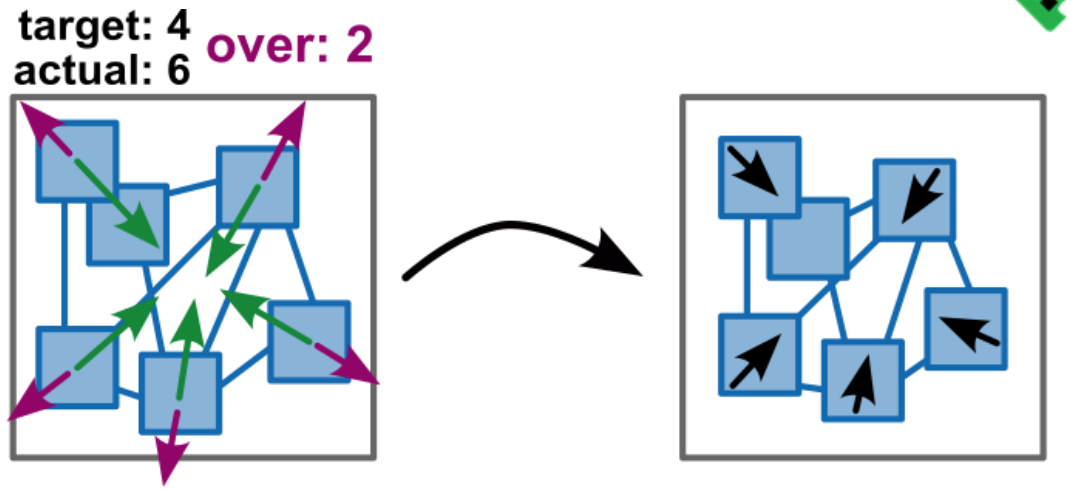
- **Highly interconnected modules**
  - More wires per cell
  - Wire-force dominates
  - Pulls cells tightly together
- **Solution: Routability driven**
  - Inflate cell area if hard to solve

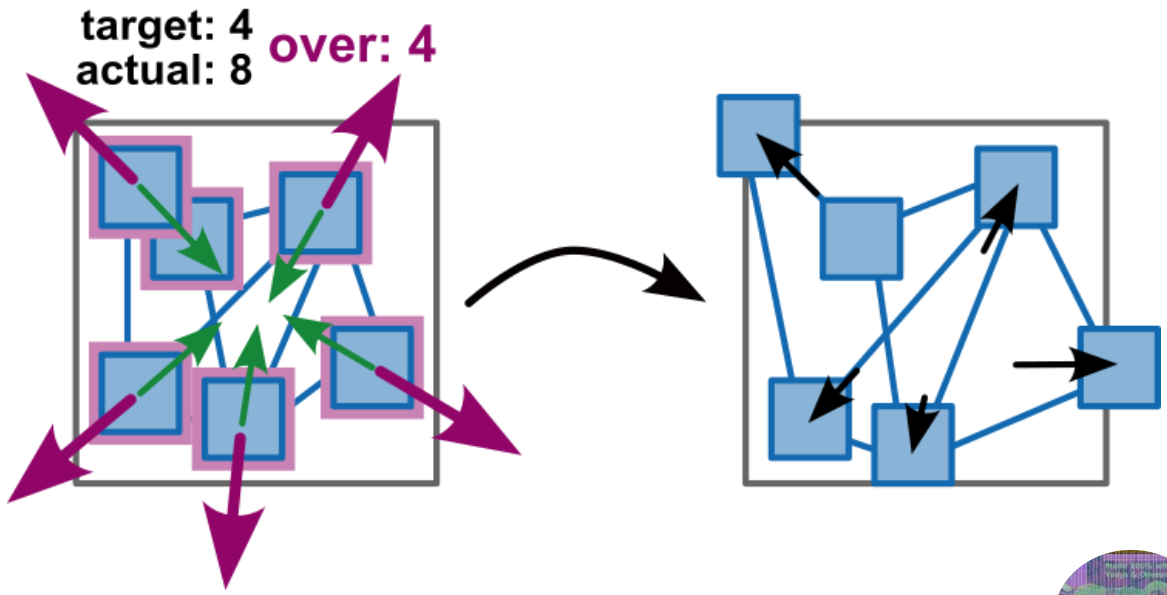




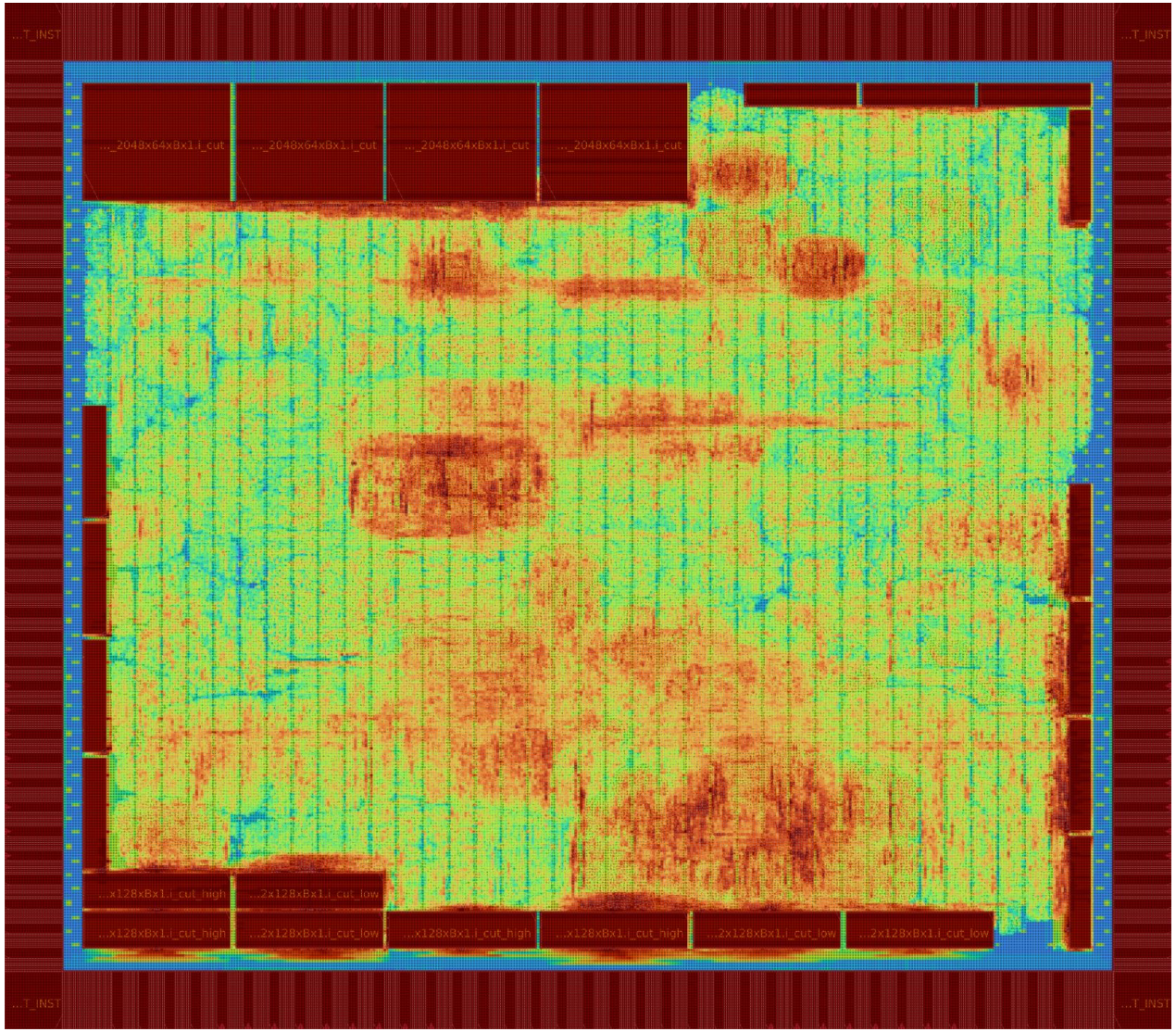
Parameter Tuning Required

- **Highly interconnected modules**
  - More wires per cell
  - Wire-force dominates
  - Pulls cells tightly together
- **Solution: Routability driven**
  - Inflate cell area if hard to solve
  - Requires manually tuning hyper-parameters
  - check_overflow: How early OR uses it
  - max_inflation_iter: Number of iterations
  - inflation_ratio_coef:
    Virtual area increase on difficult cell

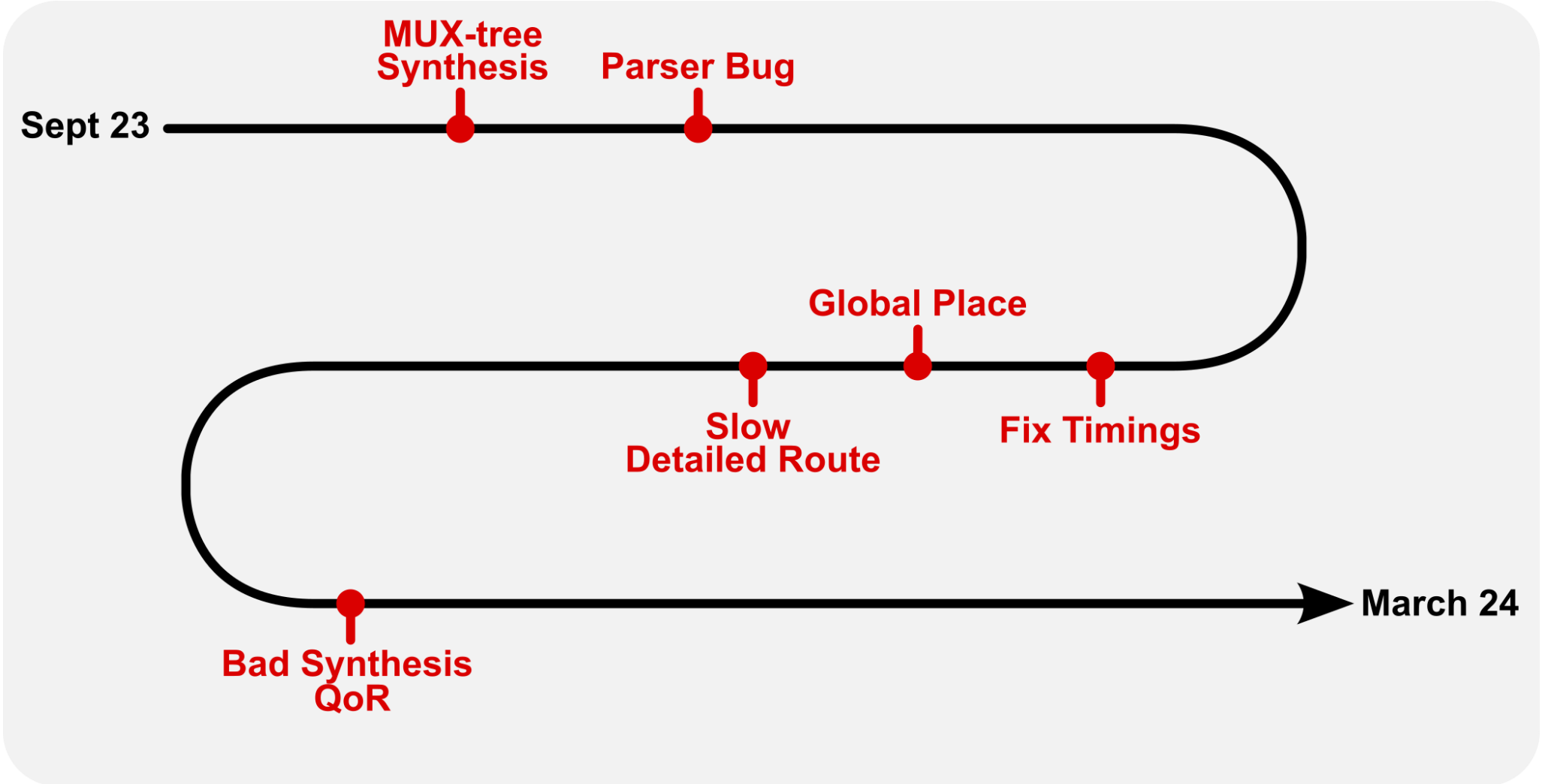




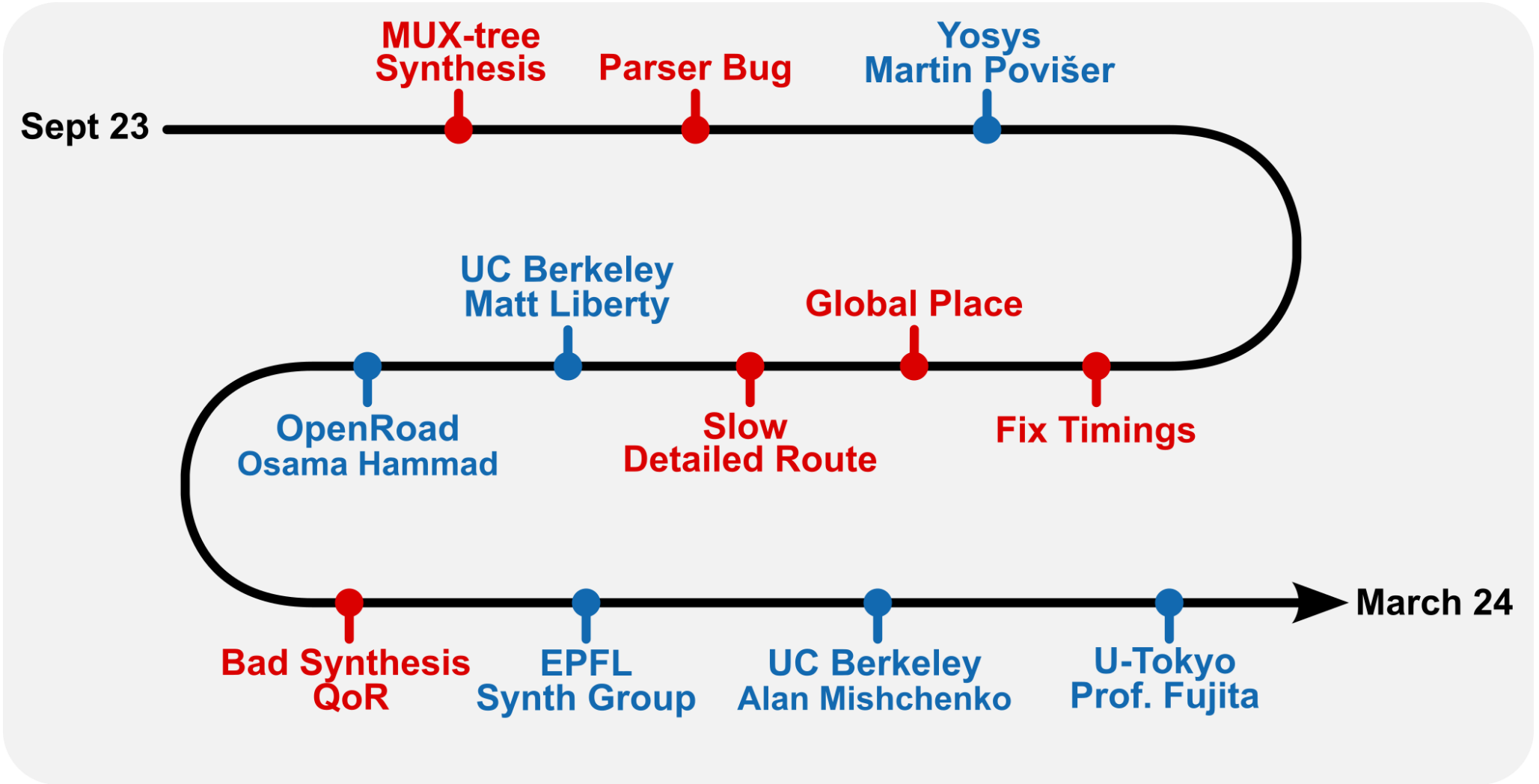
Final Placement Result



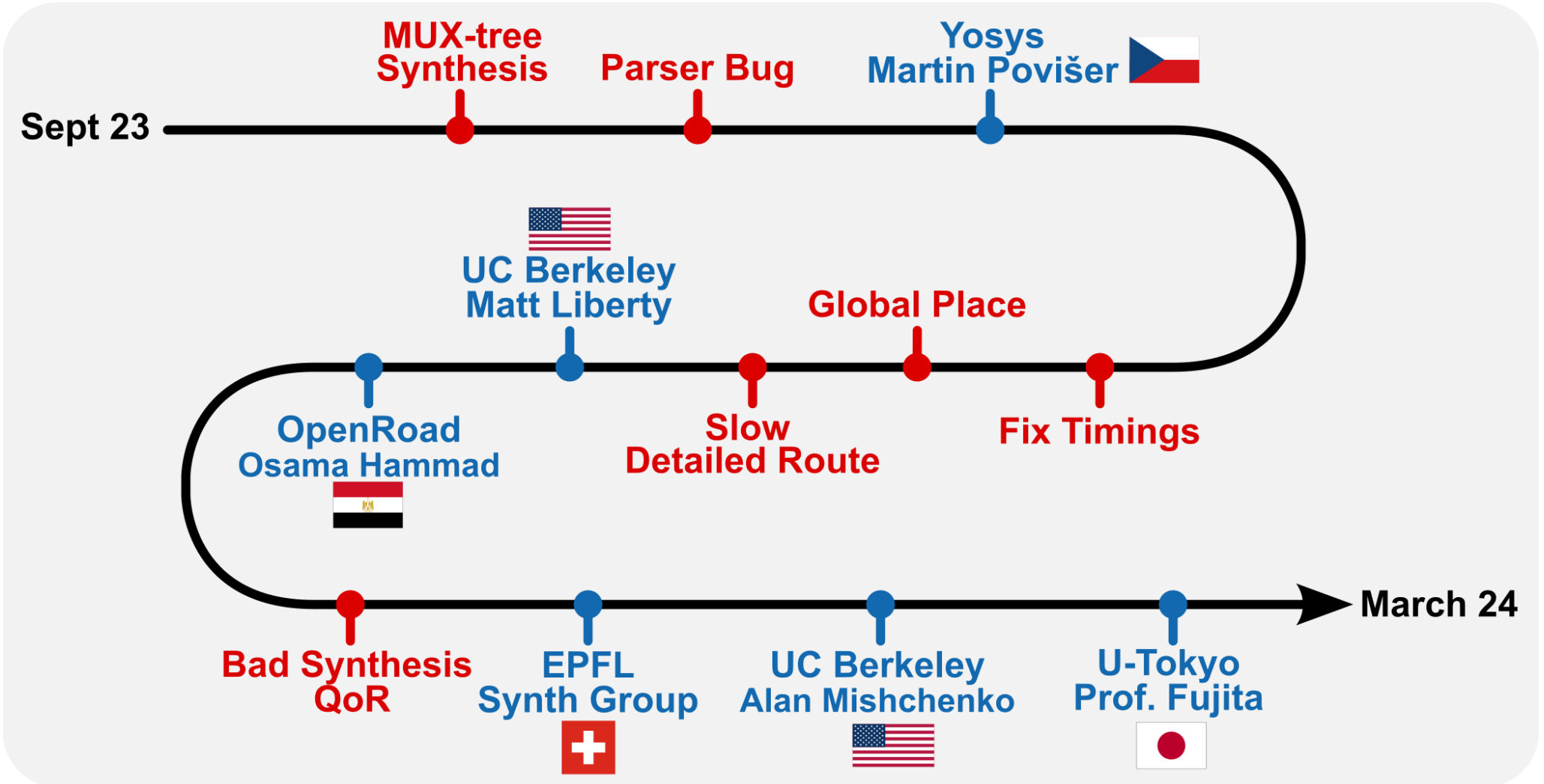
Potholes and Speed-bumps



Teamwork makes the Dream Work!



Teamwork makes the Dream Work!

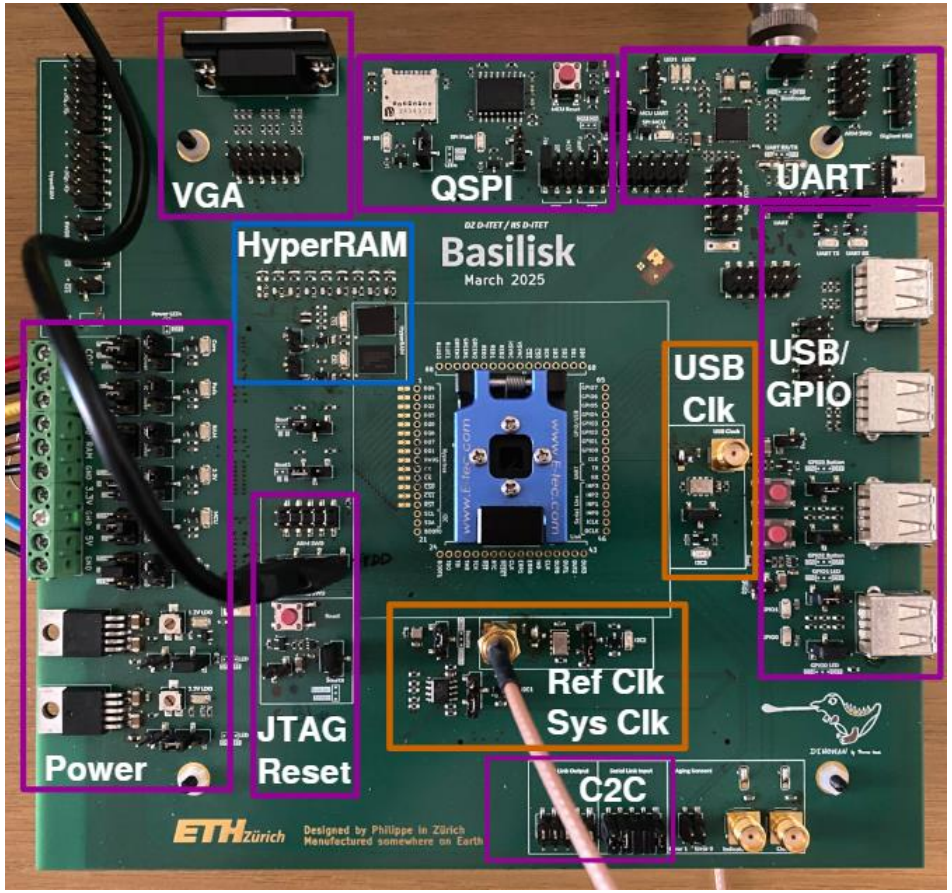


Meet Basilisk: Open **RTL**, Open **EDA**, Open **PDK**

- **Designed in IHP 130nm OpenPDK**
  - **34mm²** (6.25mm x 5.50mm)
  - **~5× larger** than previous end-to-end OS designs
  - 2.7 MGE total, 1.14MGE logic
  - 24 SRAM macros (114 KiB)
  - **62MHz** at nominal voltage (1.2V)
- **RV64GC design runs Linux**
- **Active collaboration with**

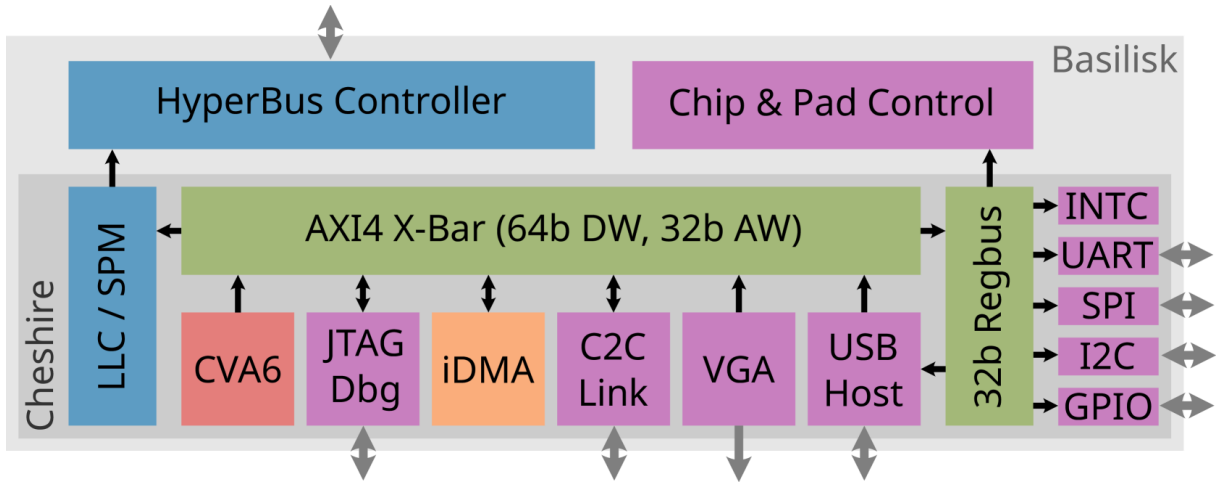
Open-source EDA tools already enable complex designs, it will only get better

Basilisk is a complete Linux-capable SoC



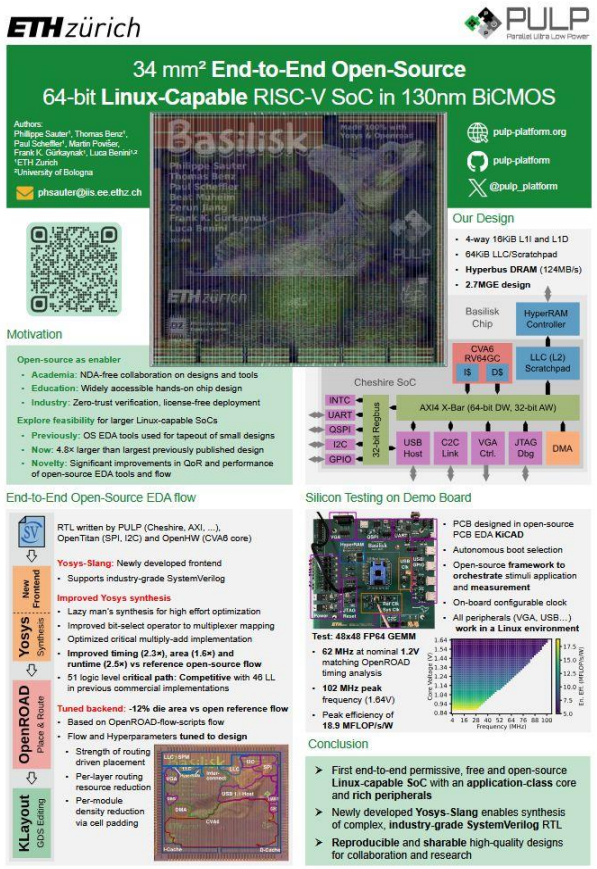
- **64-bit RISC-V core**
- **Rich peripherals:**
  - HyperRAM controller **@154MB/s**
  - C2C AXI-Link **@77MB/s**
- **Autonomous boot from SD-card**

Watch the video



arxiv.org/pdf/2505.10060

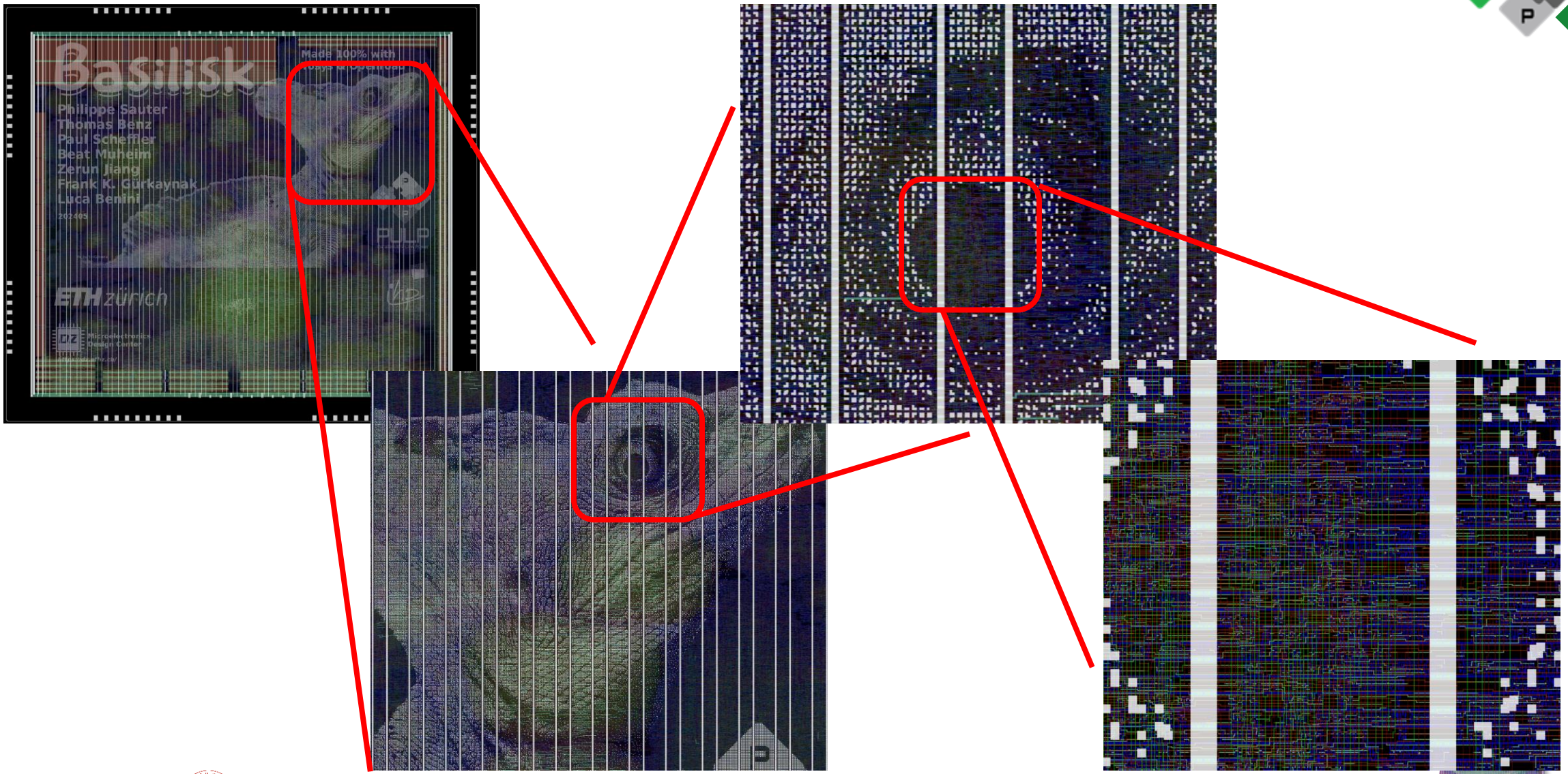
We presented Basilisk at HOT CHIPS



Poster: lnkd.in/daB6HskB

*semiwiki.com/ip/risc-v/361204-basilisk-at-hot-chips-2025-presented-ominous-challenge-to-ip-eda-status-quo/

Its Open-Source down to the Transistors



Where do we go next?

The PULP Open-Source Ecosystem

RISC-V Cores and Vector Units

RI5CY *CV32E*	Zero R *Ibex*	Snitch	Spatz	Ariane *CVA6*	ARA
RV32	RV32	RV32	RVV	RV64	RVV

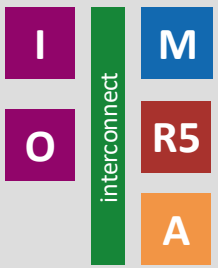
Peripherals

JTAG, SPI, UART, I2S, DMA, GPIO

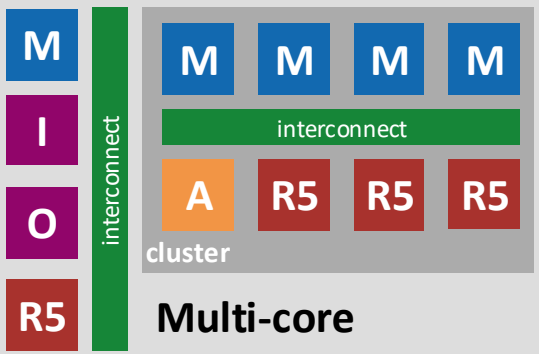
Interconnects

LIC, HCI, APB, FlooNoC, AXI4

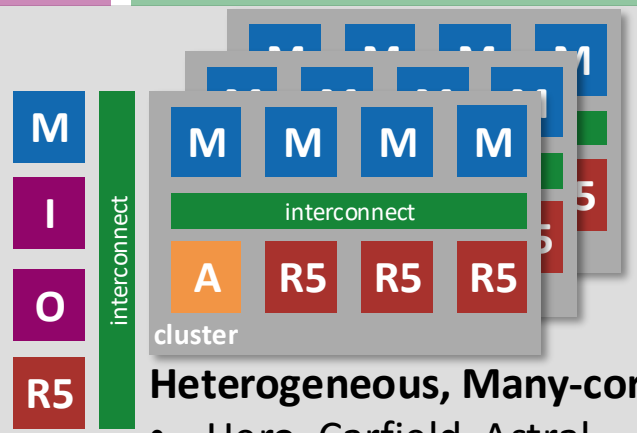
Platforms



Single core
- PULPino, PULPissimo
- **Cheshire**



Multi-core
- OpenPULP



Heterogeneous, Many-core
- Hero, Carfield, Astral
- Occamy, Mempool

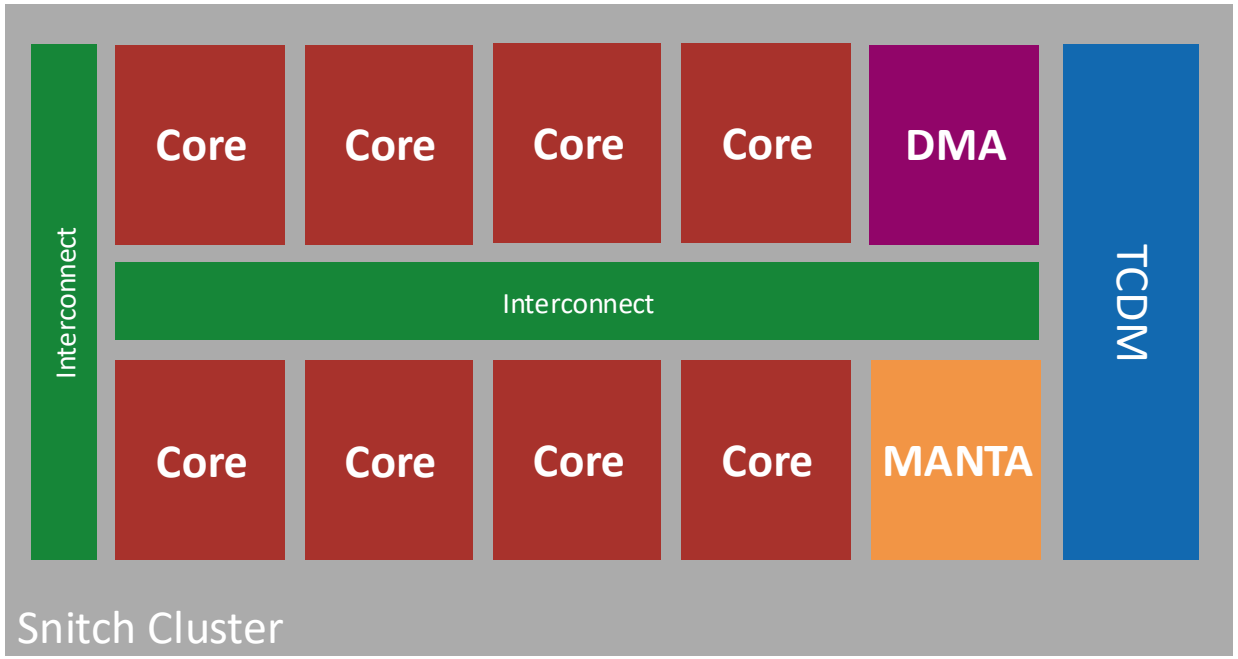
IOT → **HPC**

Accelerators and ISA extensions

XpulpNN, XpulpTNN	ITA (Transformers)	RBE, NEUREKA (QNNs)	FFT (DSP)	REDMULE (FP-Tensor)

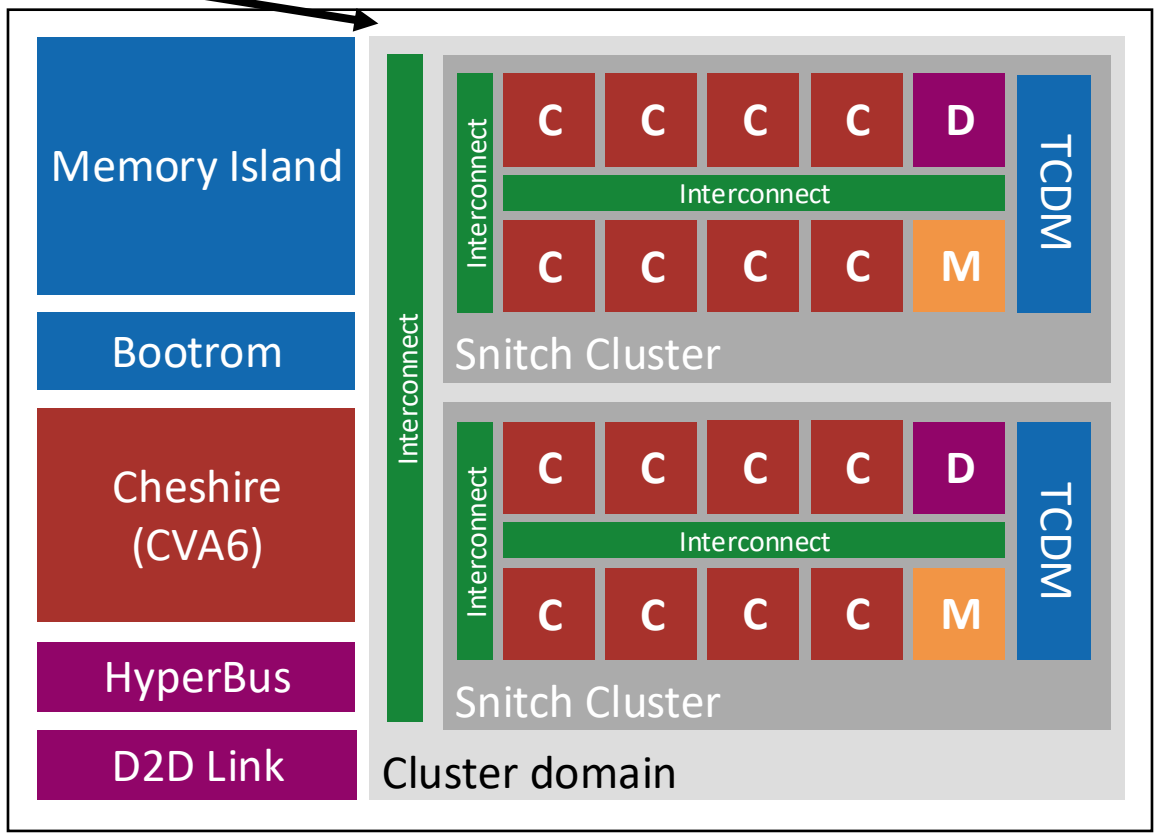
Exploring Scientific Designs with Open-Source Tools

- **Tightly Coupled L1 Memory** ■
  - Hierarchical organisation of banks
- **MANTA Accelerator** ■
  - Accelerates GEMM in MXINT8 format
  - **Main scientific contribution and focus**
- **Snitch Cores** ■
  - Optimized for data processing
  - Scalable number of cores
  - **Various extensions optimized for ML**
- **Dedicated DMA Core** ■
  - Connected to a wide AXI interface
  - Moves data to/from L2 and clusters



SeyrITA: Open-Source Tools with GF22

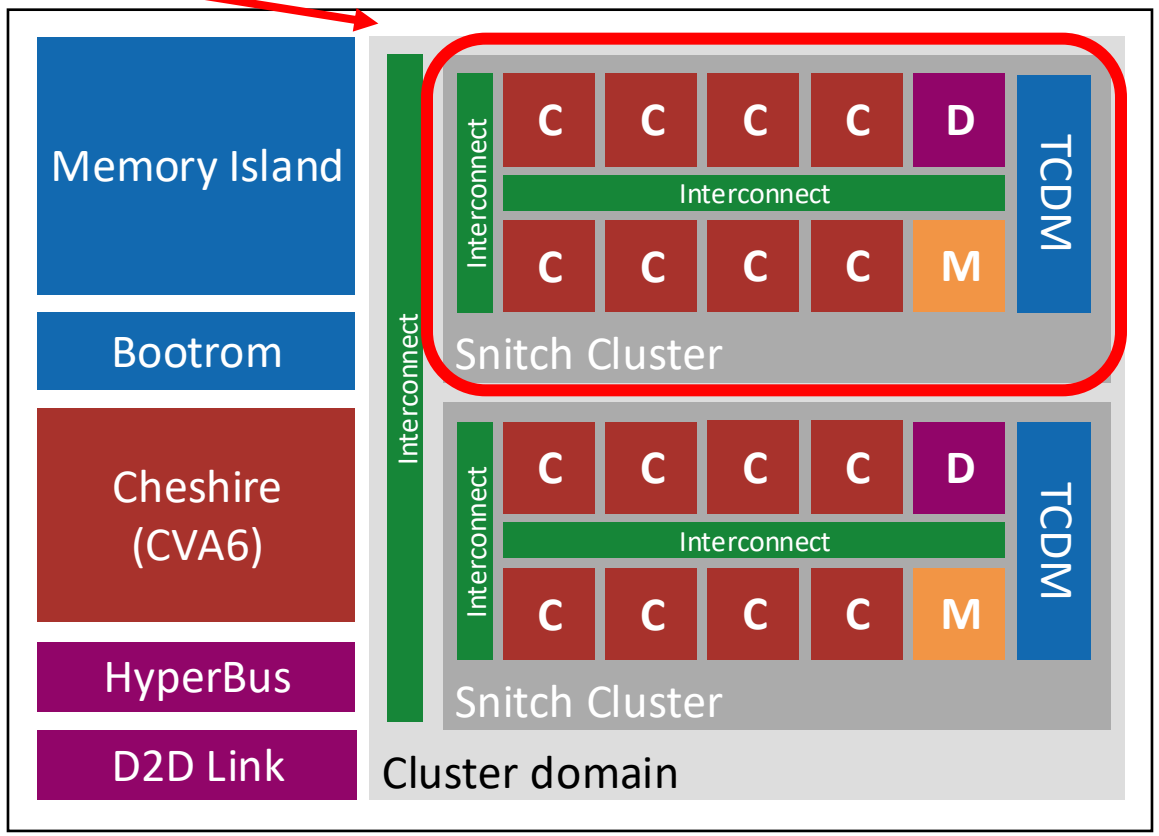
- **Transformer acceleration**
  - DeiT Image Transformer (5M Params)
  - BERT language model (4-40M Params)
  - MobileBERT-Tiny (15M Params)
  - 2-4 compute clusters
- **Memory Island**
  - ~2MB SRAM per die
- **Linux Capable Host (Cheshire)**
  - Hyperbus for DRAM
  - SD-card for persistent Flash storage
- **Die-to-die Chiplet Interconnect**
  - 4Gbit/s per lane, 2-4 lanes



SeyrITA: Open-Source Tools with GF22

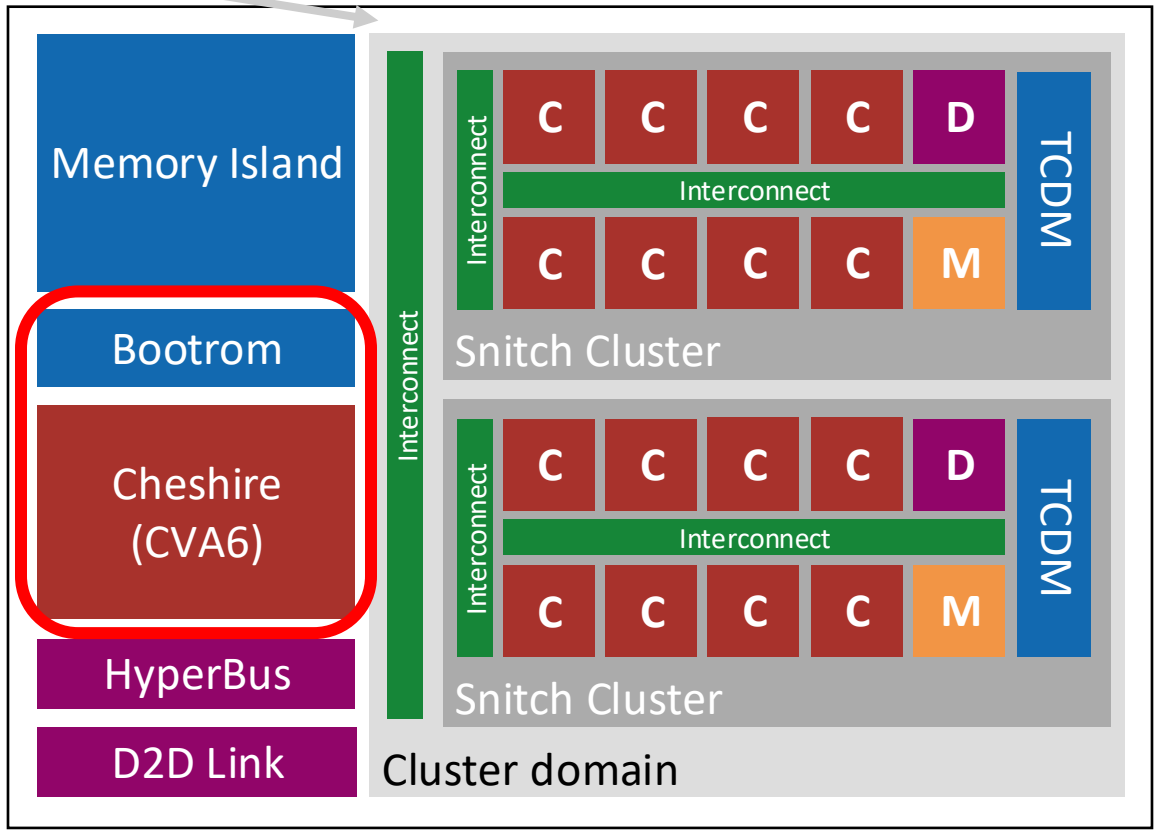
- **Transformer acceleration**
  - DeiT Image Transformer (5M Params)
  - BERT language model (4-40M Params)
  - MobileBERT-Tiny (15M Params)
  - 2-4 compute clusters
- Memory Island
- ~2MB SRAM per die
- Linux Capable Host (Cheshire)
  - Hyperbus for DRAM
  - SD-card for persistent Flash storage
- Die-to-die Chiplet Interconnect
  - 4Gbit/s per lane, 2-4 lanes

The focus is on the Transformer Acceleration



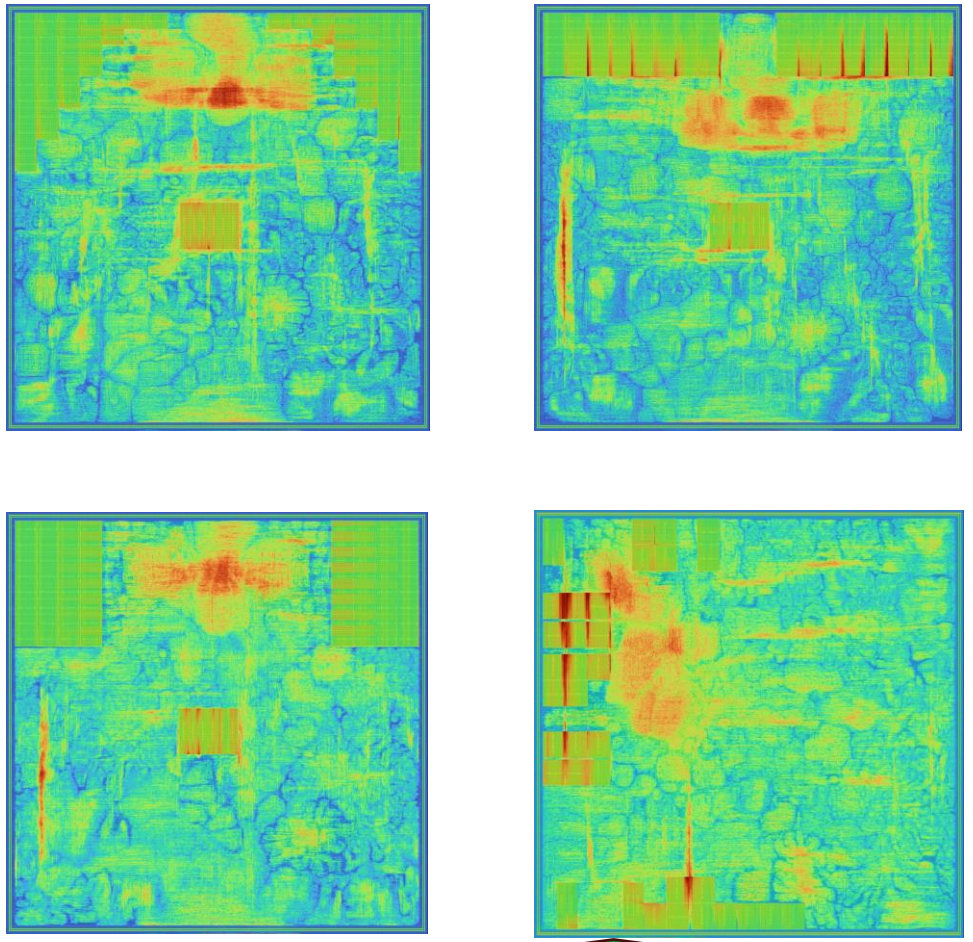
SeyrITA: Open-Source Tools with GF22

- **Transformer acceleration**
  - DeiT Image Transformer (5M Params)
  - BERT language model (4-40M Params)
  - MobileBERT Tiny (15M Params)
- **Basilisk system as a small part**
  - 2-4 compute clusters
  - **SeyrITA is 10-15x Basilisk!**
  - Memory Island
- **~2MB SRAM per die**
- **Linux Capable Host (Cheshire)**
  - Hyperbus for DRAM
  - SD-card for persistent Flash storage
- **Die-to-die Chiplet Interconnect**
  - 4Gbit/s per lane, 2-4 lanes



On the Horizon: SeyrITA – GF22 with Open-Source Tools

- **Demonstrate** a large **22nm tapeout** with open-source tools
- **Improve tools** and close the **performance** gap
- Identify and **implement missing features** along the way
- **Active Collaboration with**



Snitch cluster floorplan exploration

Benefits of end to end openness

Research

- Easier collaboration (no NDAs)
- Reproducible results, benchmarking
- **Combined impact of design and design automation**

Industry

- Transparent chain of trust, sovereignty
- Lower initial cost
- **Accelerate research to product pipeline**

Education

- Increased accessibility
- No black boxes, full visibility
- **Experiment with flows and tools**

Any Questions?

- **We use open source because it works**
  - Allows us to manage complex designs
  - Facilitates Industry/Academia Relationships
  - Creates Auditable Designs, Reproducible Results
  - Enables research into new directions

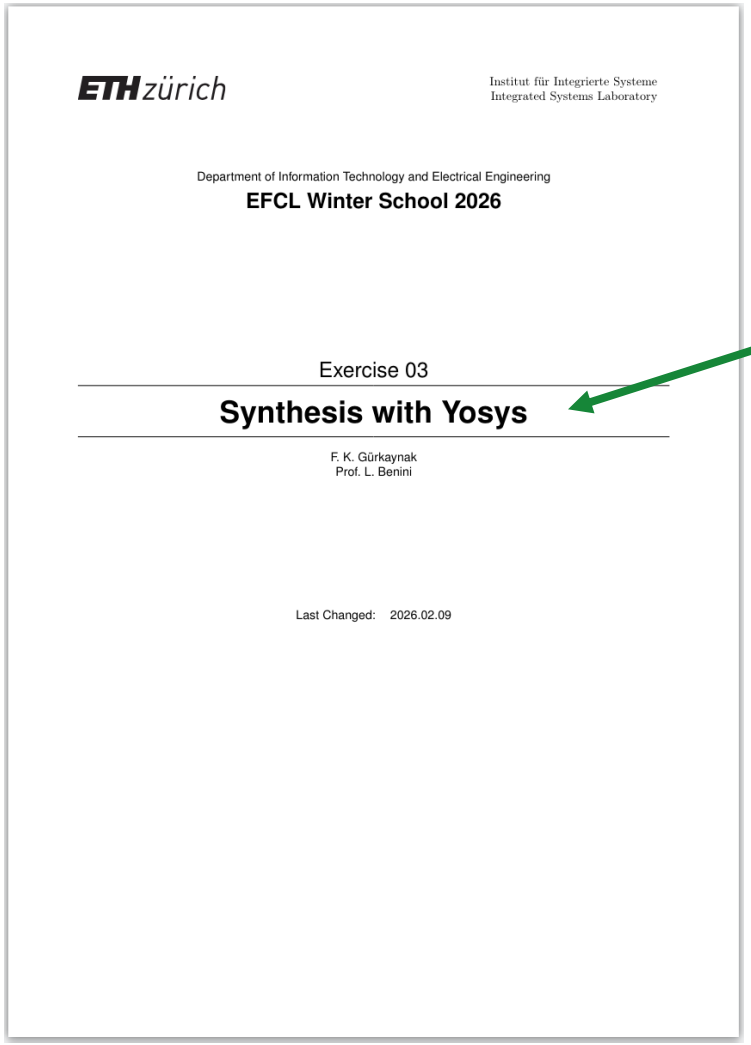
There is still more to come

Helps us and others concentrate work where it matters

- **Open Source sees no borders**
  - There is **no** ‘European/Chinese/American Open Source’,
  - There **can be** ’European/Chinese/American support for Open Source’

Open Source is global, it just can have more or less support in a region/country

Next up: Intro to Synthesis



ETH zürich

Institut für Integrierte Systeme
Integrated Systems Laboratory

Department of Information Technology and Electrical Engineering

EFCL Winter School 2026

Exercise 03

Synthesis with Yosys

F. K. Gürkaynak
Prof. L. Benini

Last Changed: 2026.02.09

There is still more to come

For you that's the Synthesis exercise☺