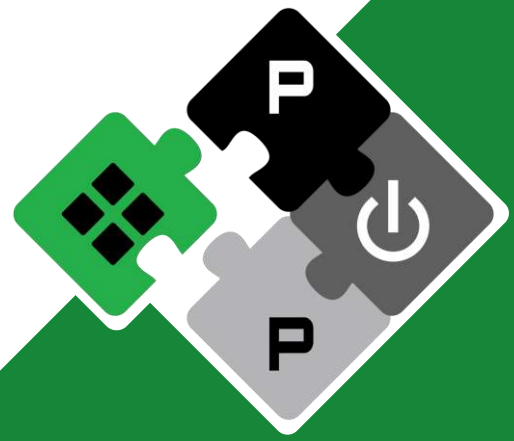
ETH zürich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Designing Linux-capable systems using open EDA tools

Integrated Systems Laboratory (ETH Zürich)

Philippe Sauter phsauter@iis.ee.ethz.ch



@pulp_platform

pulp-platform.org

youtube.com/pulp_platform

PULP Platform

Open Source Hardware, the way it should be!

PULP Platform by ETH Zürich and University of Bologna

OCCAMY

432 RISC-V cores

Chiplets

GF12nm

1GHz

2024 VLSI SYMPOSIUM

Occamy: A 432-Core 28.1 DP-GFLOP/s/W 83% FPU Utilization Dual-Chiplet, Dual-HBM2E RISC-V-based Accelerator for Stencil and Sparse Linear Algebra Computations with 8-to-64-bit Floating-Point Support in 12nm FinFET

...er,* Manuel Eggimann,*
...arco Ottavi,‡

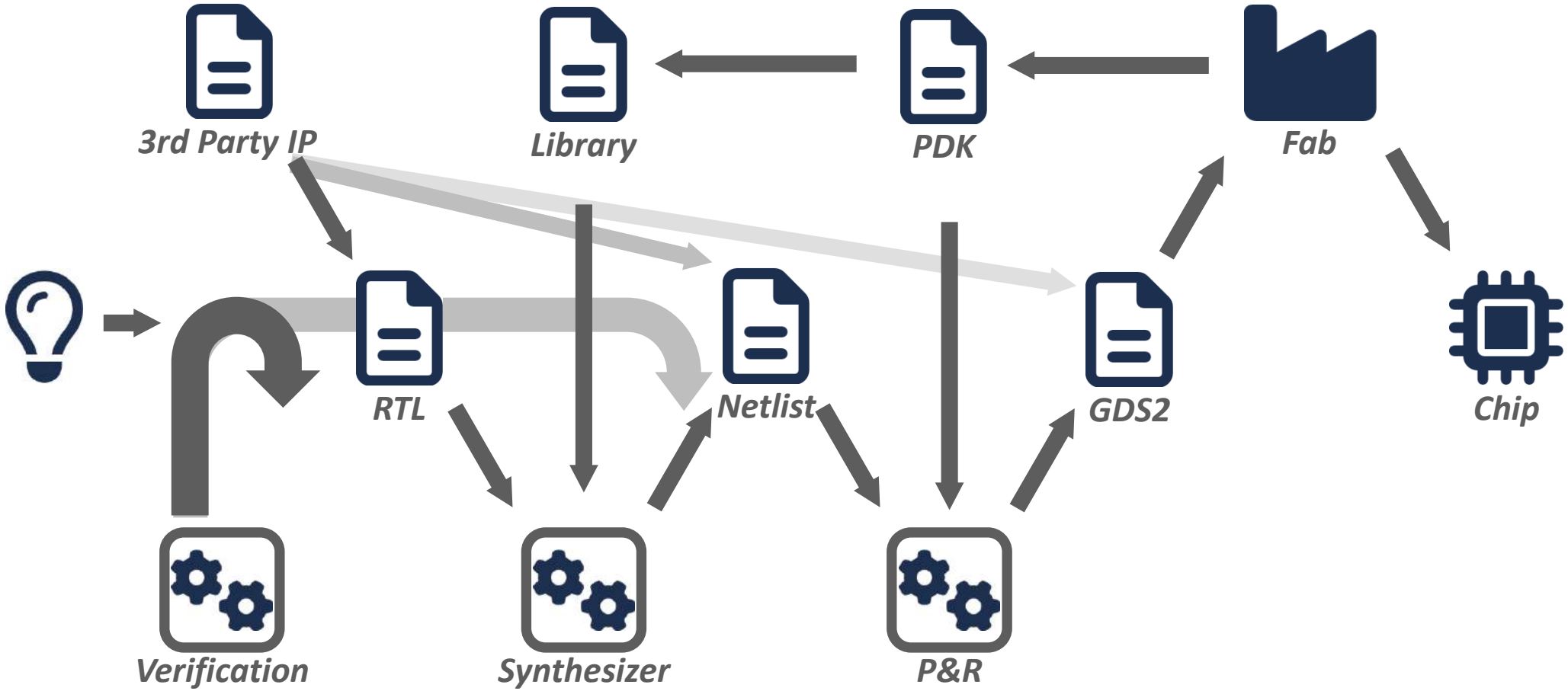
How do we manage to design projects of this size at a University?

GlobalFoundries Rambus Micron SYNOPSYS AVERY design systems

In 11 years PULP team has designed more than 60 chips

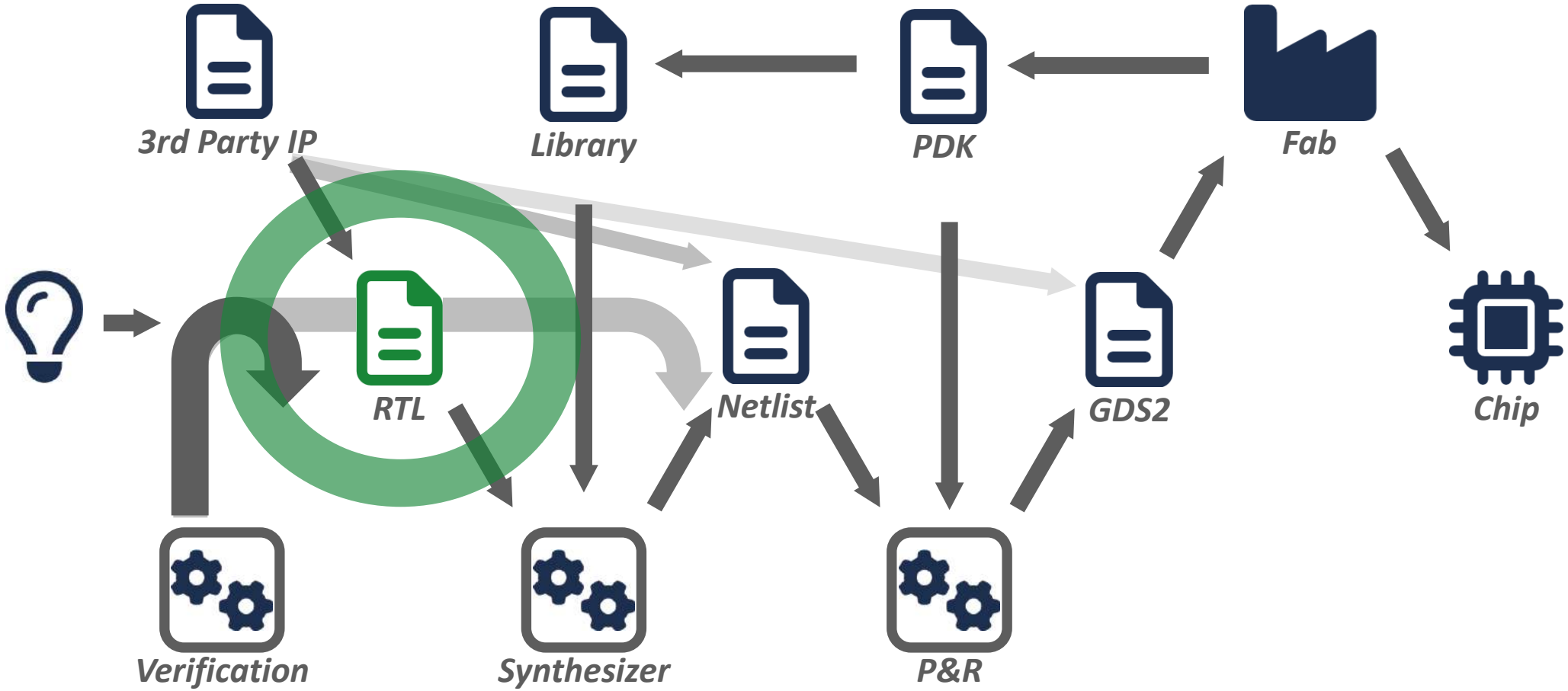
RISC-V and open-source hardware have been instrumental in our success

A simplified view of the IC design flow



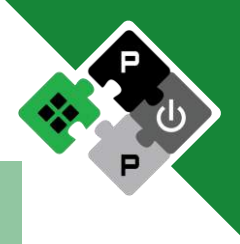
Icons taken from free icons from fontawesome.com

Most of open source hardware is at RTL level



Icons taken from free icons from fontawesome.com

We have created a sandbox to design System on Chips



RISC-V Cores and Vector Units

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

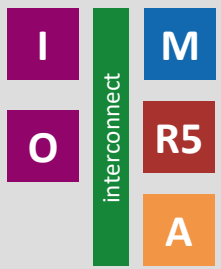
Peripherals

JTAG, SPI, UART, I2S, DMA, GPIO

Interconnects

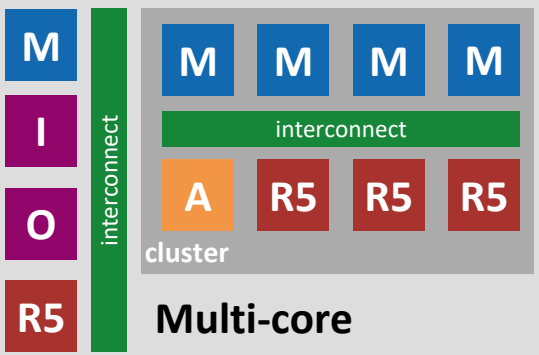
LIC, HCI, APB, FlooNoC, AXI4

Platforms



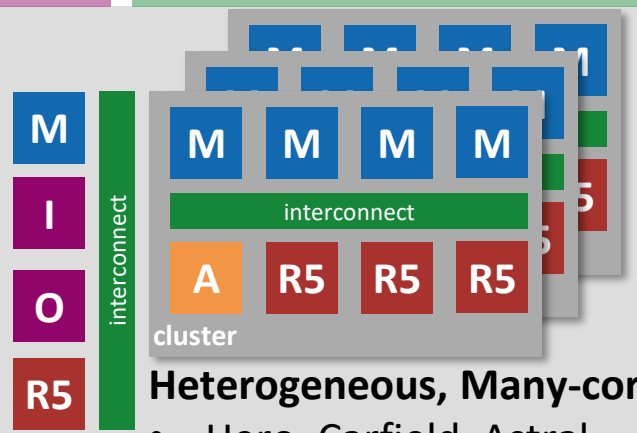
Single core

- PULPino, PULPissimo
- Cheshire



Multi-core

- OpenPULP
- ControlPULP



Heterogeneous, Many-core

- Hero, Carfield, Astral
- Occamy, Mempool

IOT → **HPC**

Accelerators and ISA extensions

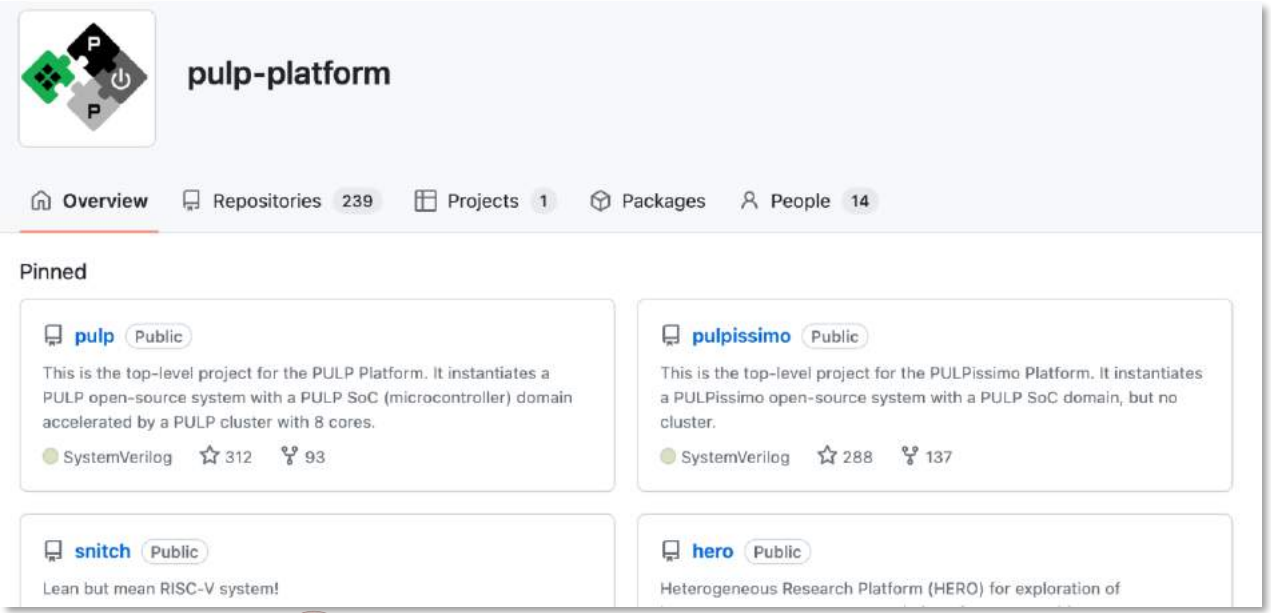
XpulpNN, XpulpTNN	ITA (Transformers)	RBE, NEUREKA (QNNs)	FFT (DSP)	REDMULE (FP-Tensor)

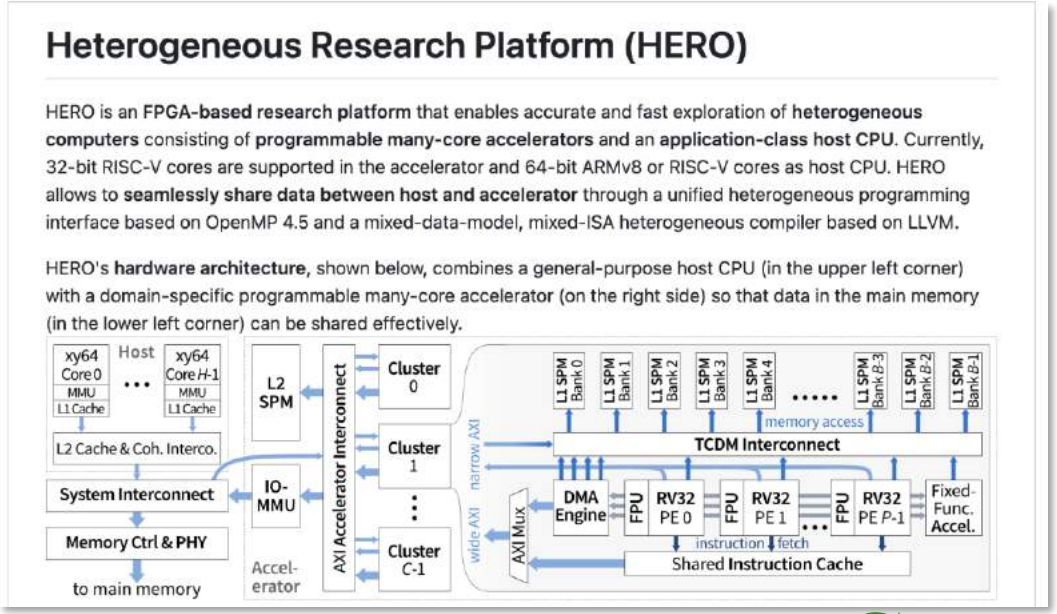
We make everything (we can) available openly

- **All our development is on GitHub using a permissive license**
  - HDL source code, testbenches, software development kit, virtual platform

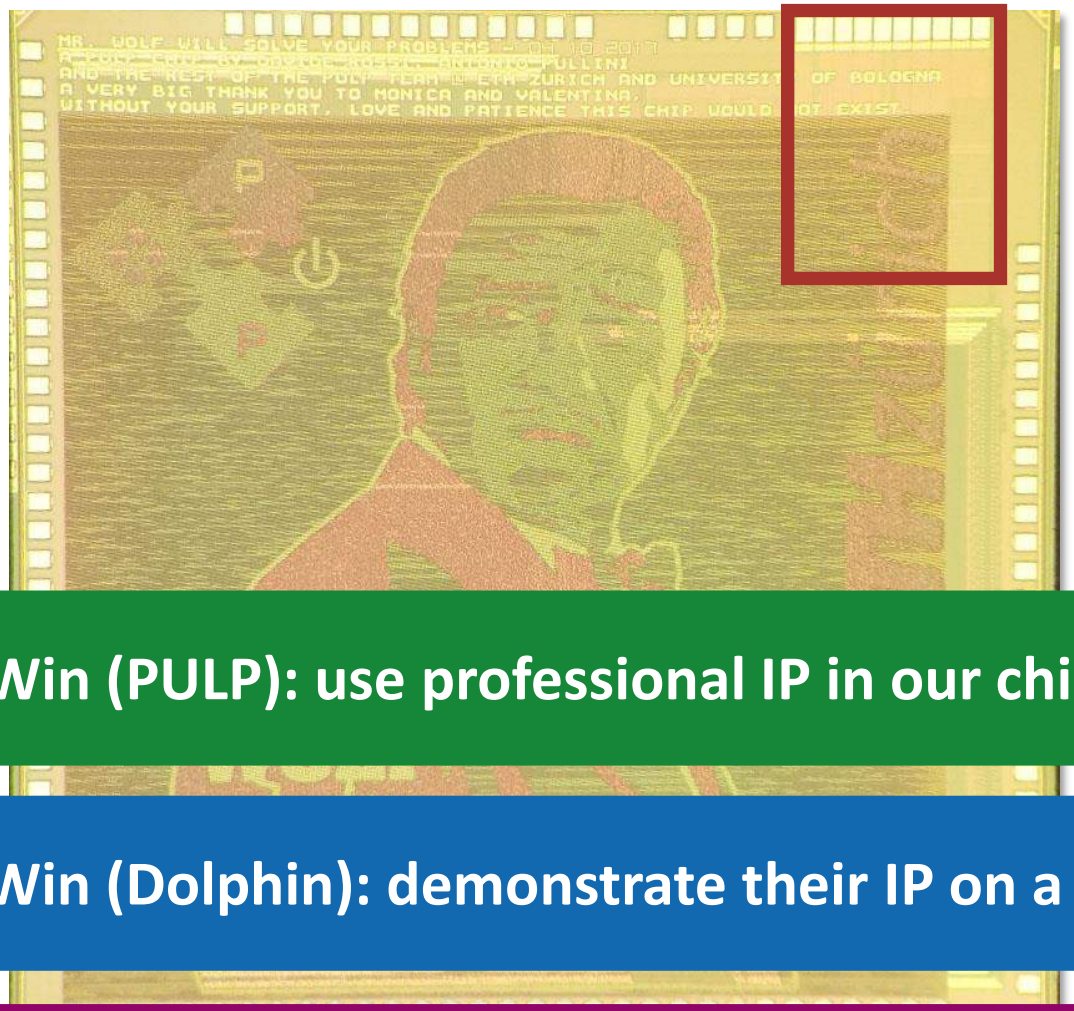
https://github.com/pulp-platform

- Allows anyone to use, change, and make products without restrictions.





Meet Mr. Wolf (2017) in TSMC40



- **Very successful IoT processor**
  - 8+1 RISC-V cores
- **Power converter IP from Dolphin**
  - Allowed the company to demonstrate their IP on a industry relevant design
  - RTL for the entire SoC openly available
- **Design formed the basis of GAP8/9**
  - By Greenwaves Technologies

Win (PULP): use professional IP in our chips

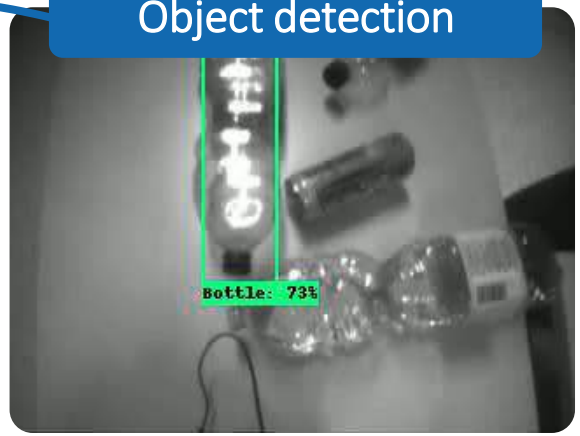
Win (Dolphin): demonstrate their IP on a SoA design

Win (Greenwaves): SoC template that can be easily productized

Designs derived from Mr. Wolf powered our nano-drones

Multiple, complex, heterogeneous tasks at high speed and robustness using **only our onboard RISC-V based IoT processors**

Object detection

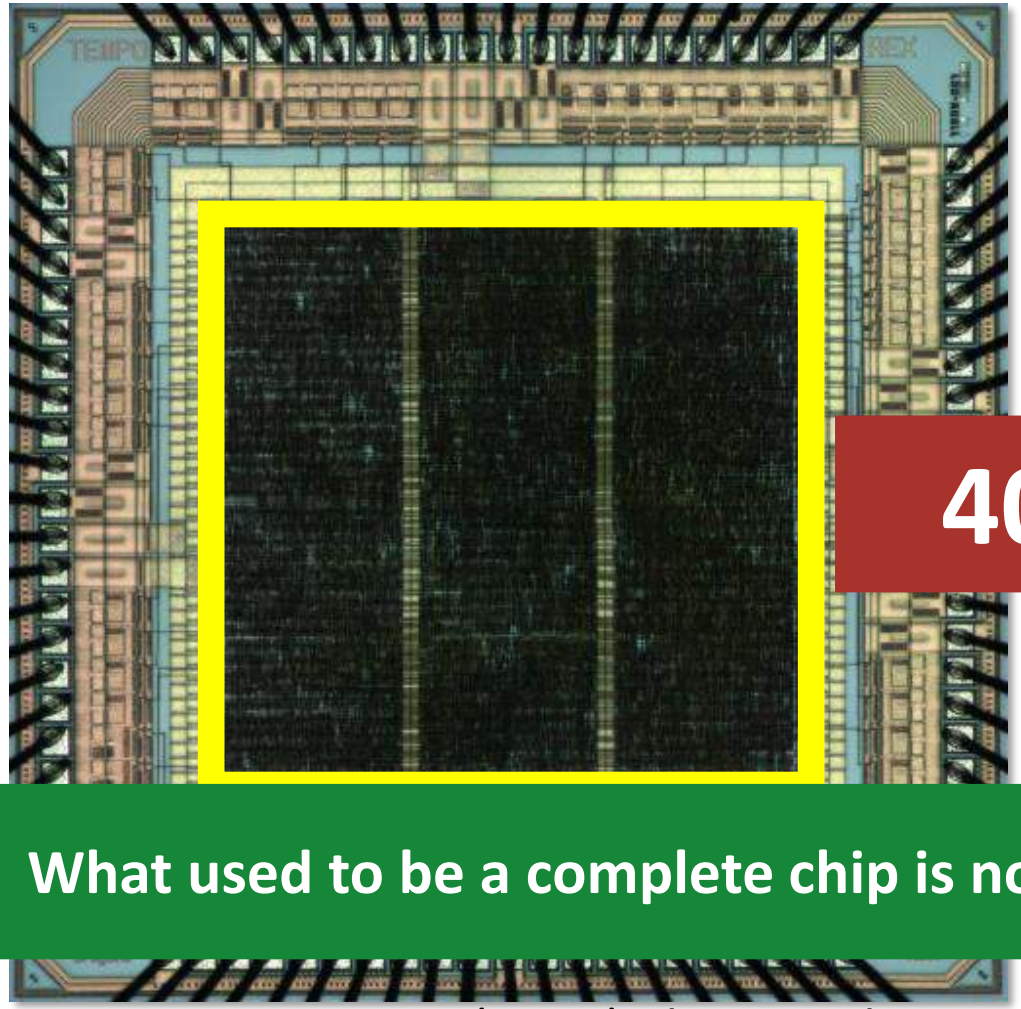


Obstacle avoidance & Navigation

Environment exploration

Multi-GOPS workload at extreme efficiency → P_{max} 100mW

In the last 20 years IC Design has changed a lot



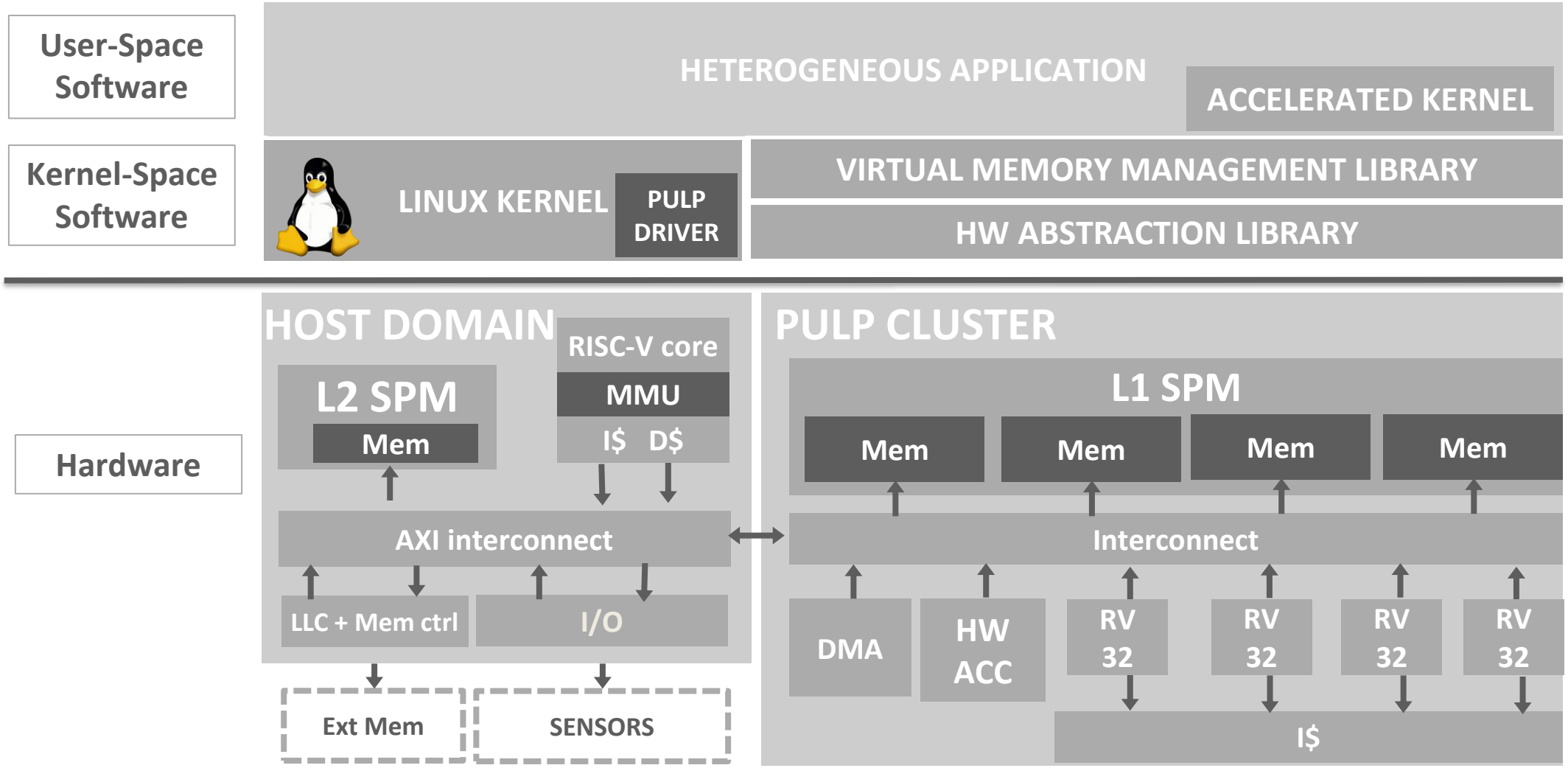
4000 x

What used to be a complete chip is now a small part of a SoC !

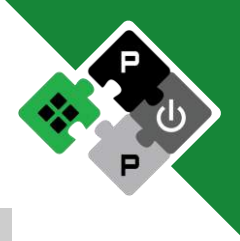
Temporex AMS 0.6 (2001) about 20 kGE

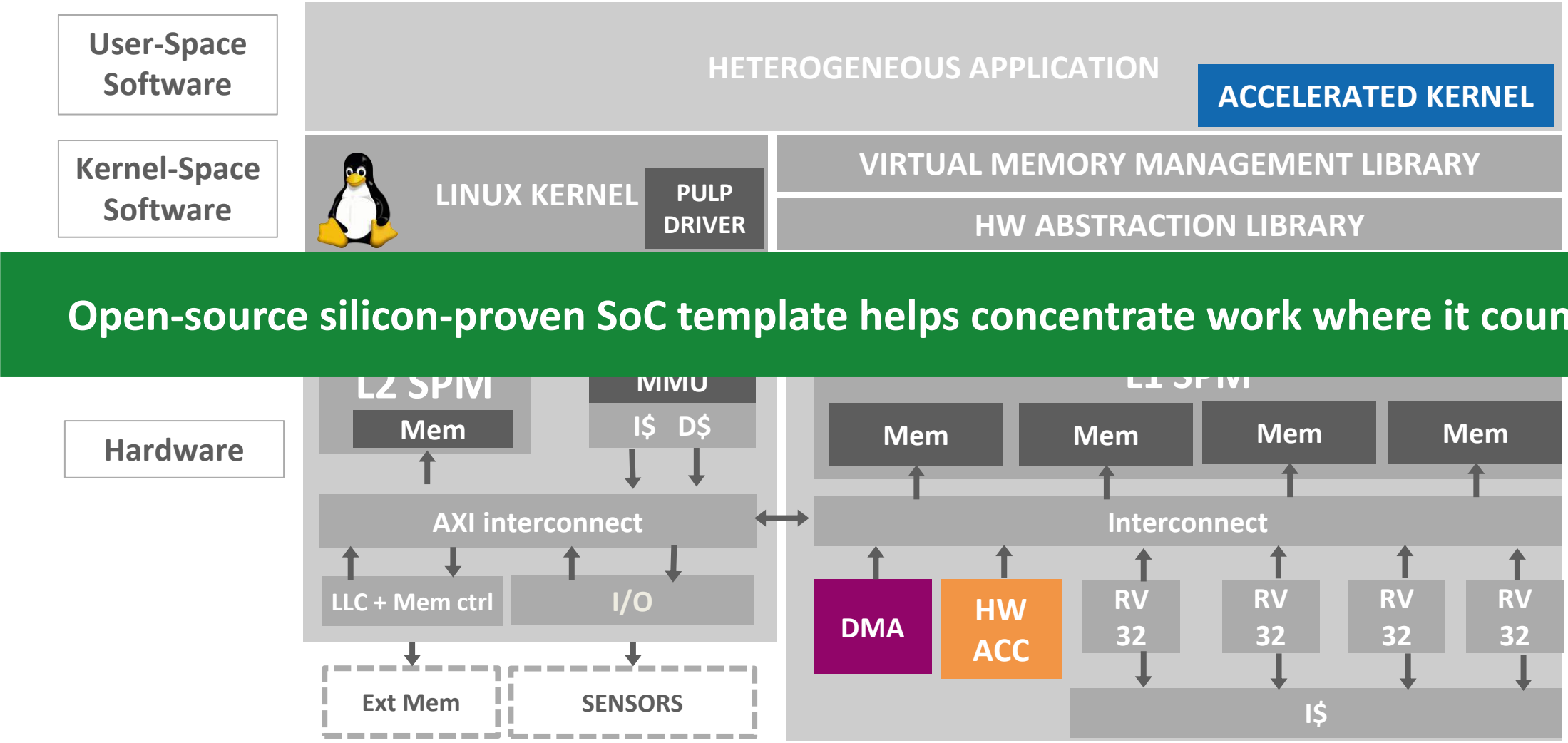
Kraken GF22 (2021) about 80 MGE

There is so much that makes up a modern SoC



In a typical design, innovation is only in a limited scope





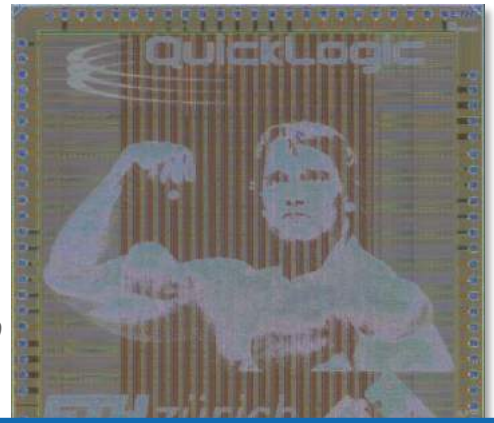
Diverse set of open source based industry collaborations

GF22 (2018)

Arnold

eFPGA coupled with a RISC-V microcontroller.

In one year from agreement to actual tapeout

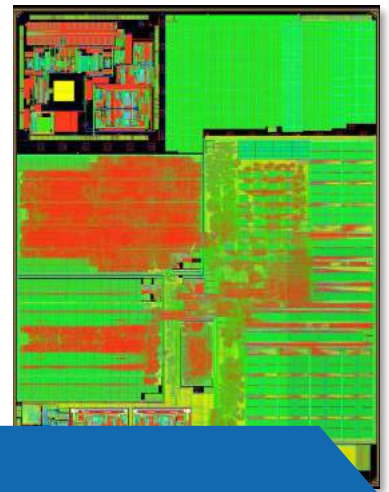


GF22 (2022)

Marsellus

Heterogeneous IoT processor With Aggressive voltage scaling

DOLPHIN

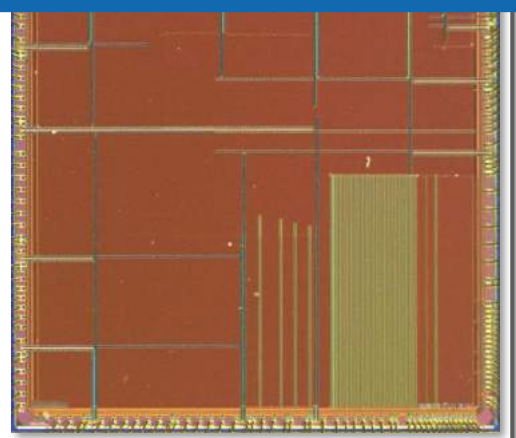


Permissive open-source licensing key to our industrial relationships

Siracusa

SoC for Extended Reality visual processing

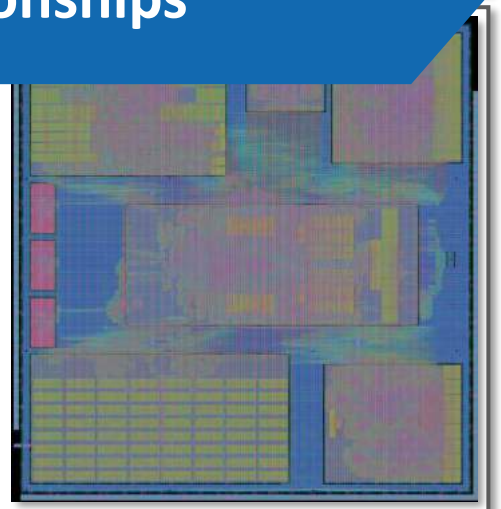
Meta



Carfield

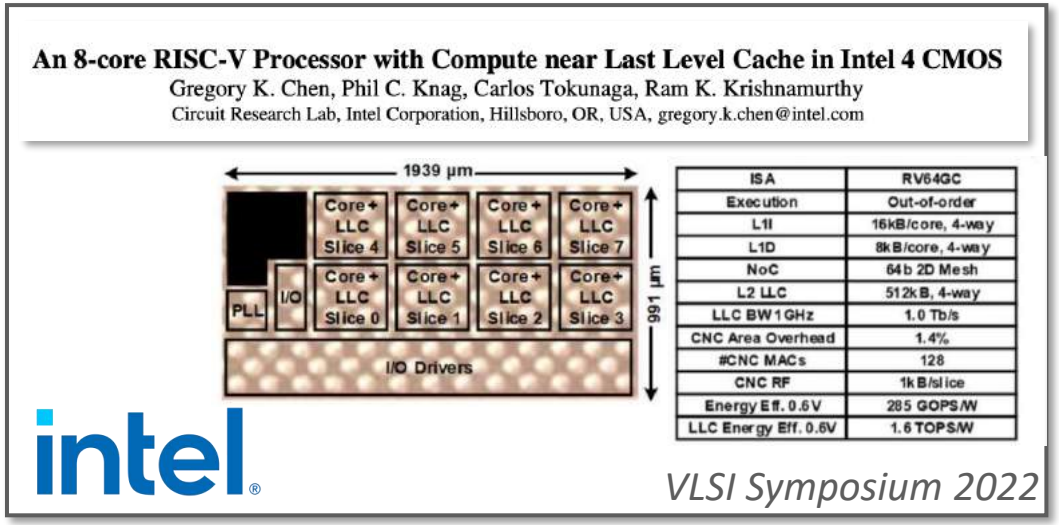
Open-Research platform for safety, resilient and time-predictable systems

intel



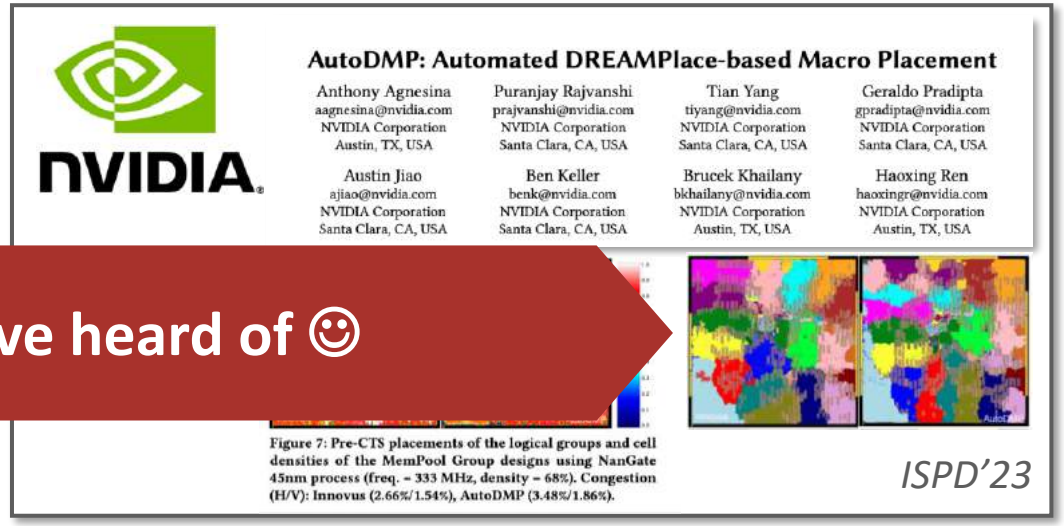
And many continue to use our work for their research

RISC-V week Barcelona 2018



VLSI Symposium 2022

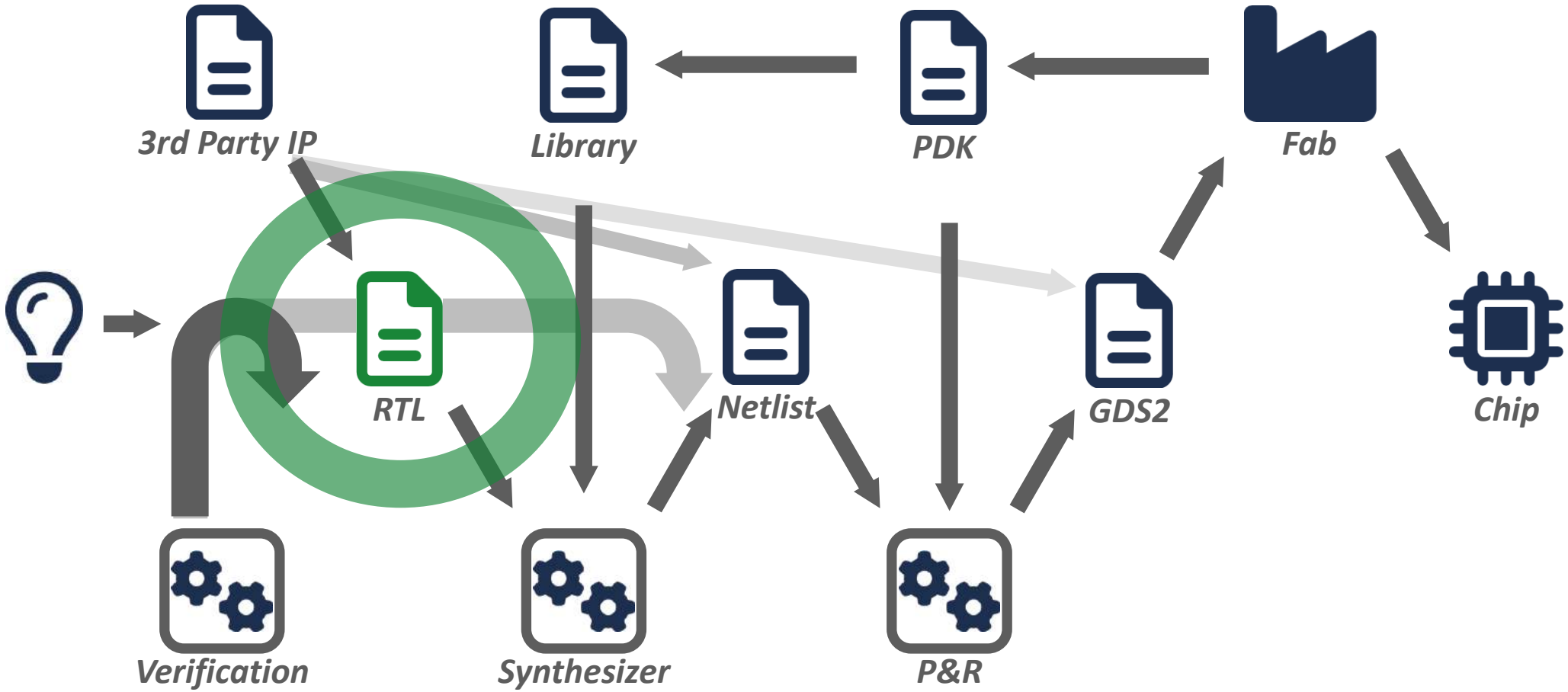
ISSCC Keynote 2020 – Nature 2020



ISPD'23

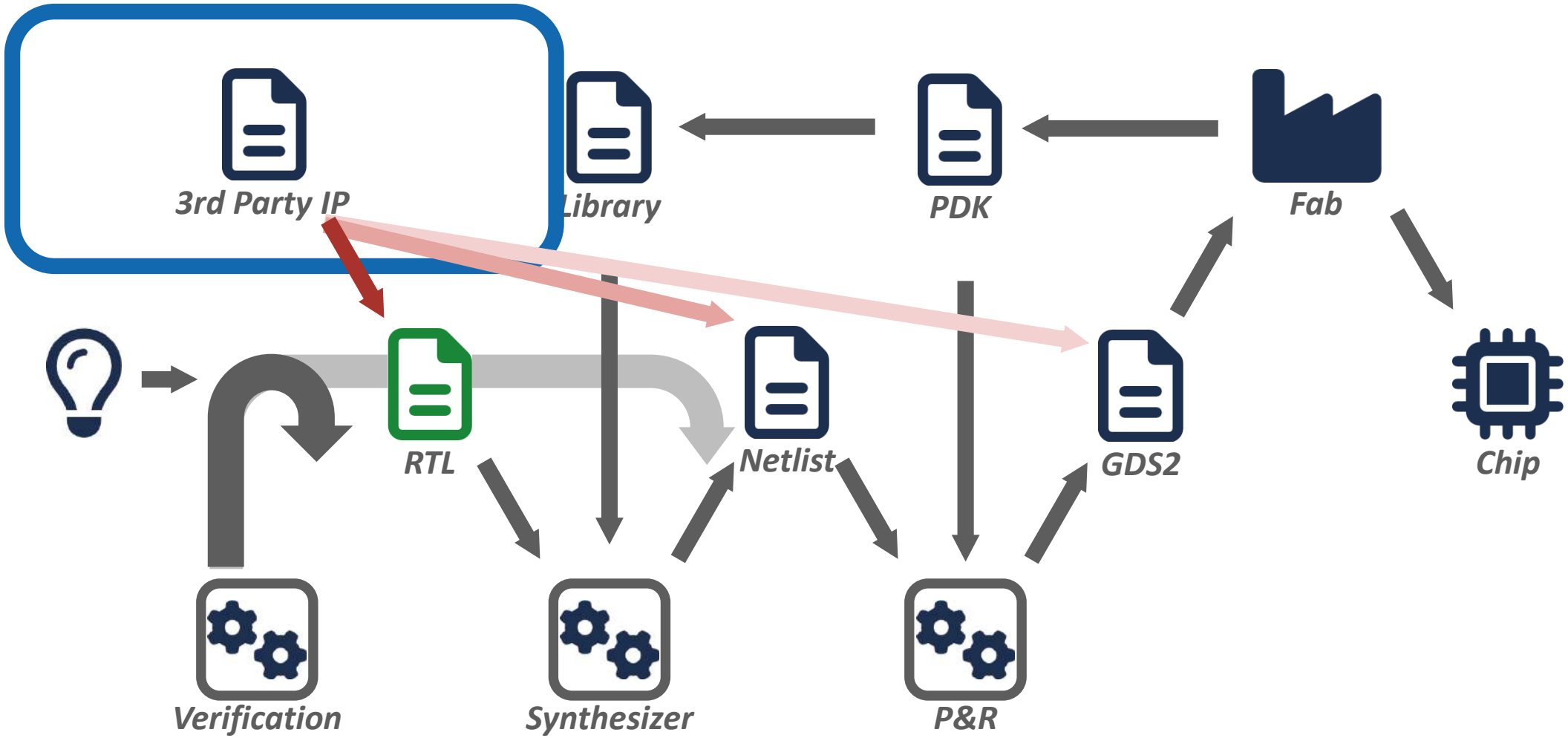
Some smaller companies you might have heard of ☺

Unlocking the rest of the design flow



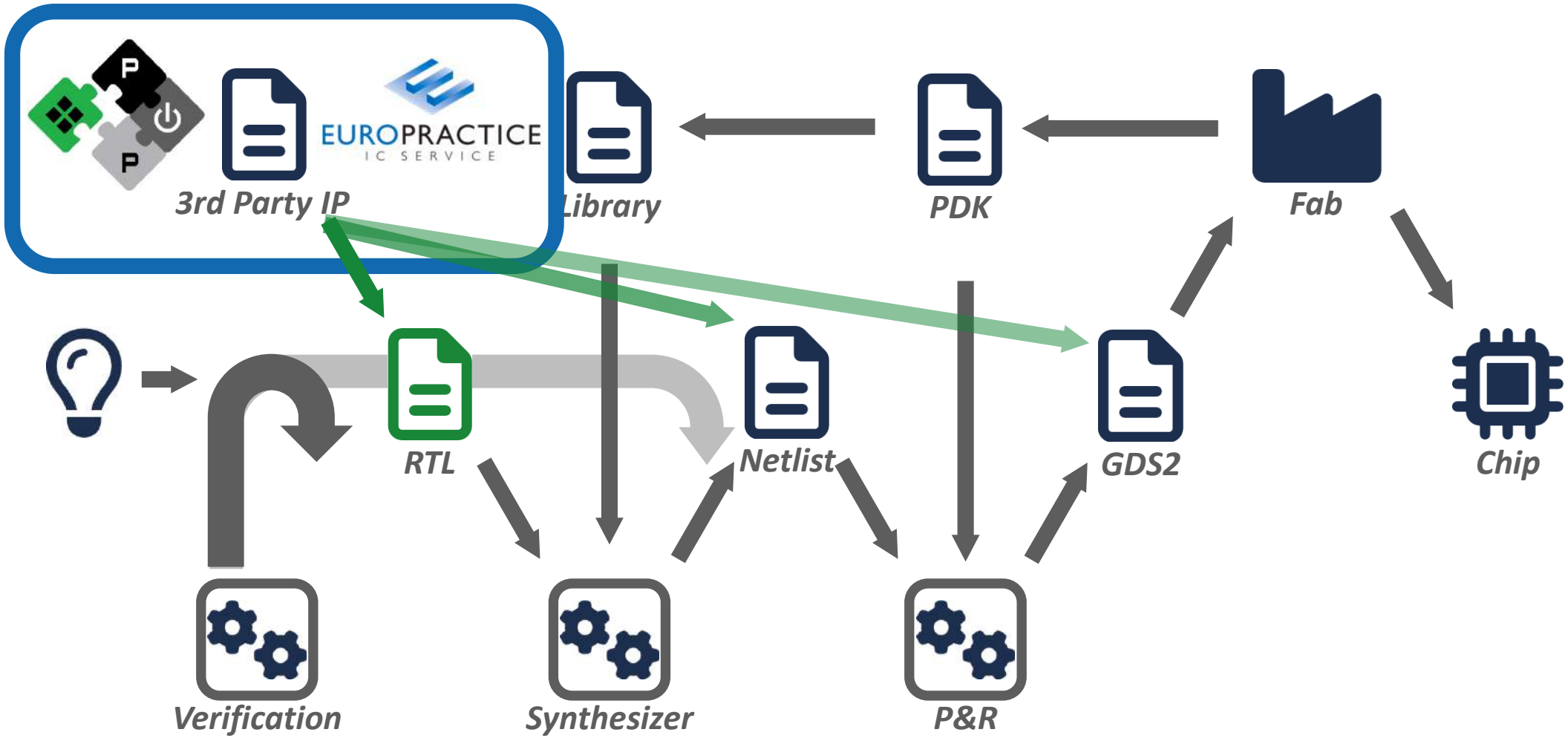
Icons taken from free icons from fontawesome.com

Most designs will include some 3rd party IP



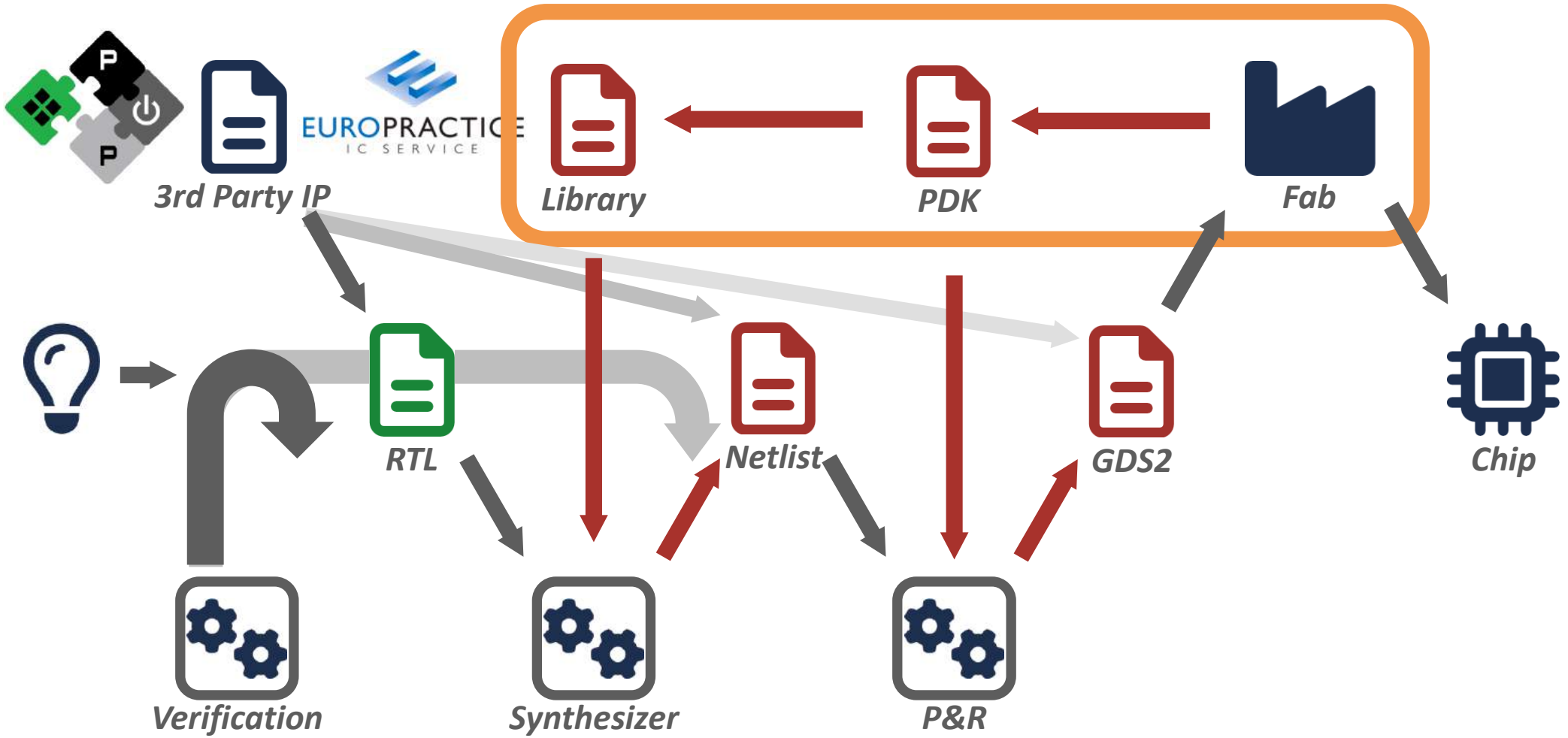
3rd party IP when included can limit what can be open sourced

Most designs will include some 3rd party IP



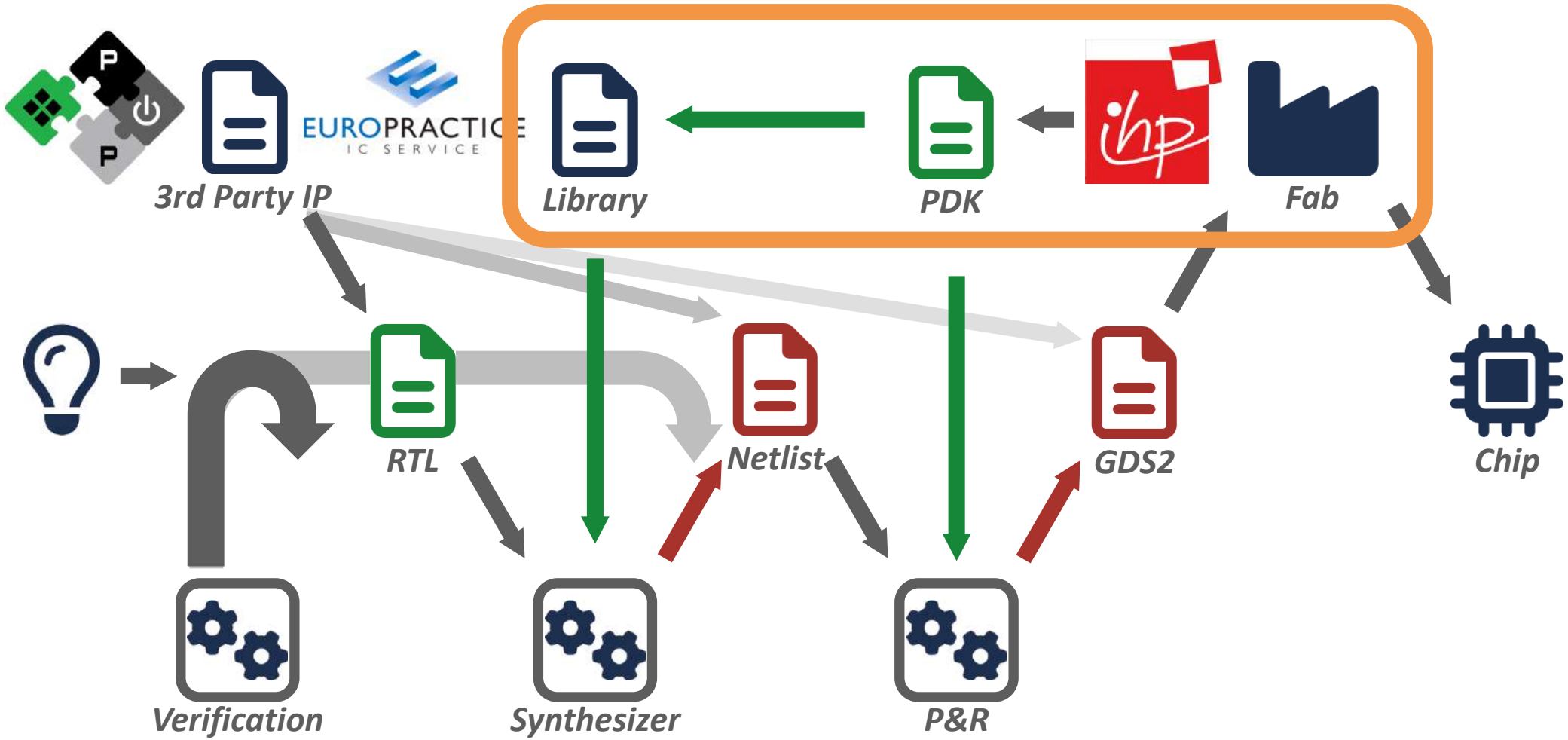
Icons taken from free icons from fontawesome.com

The chip will contain information from the PDK of the Fab



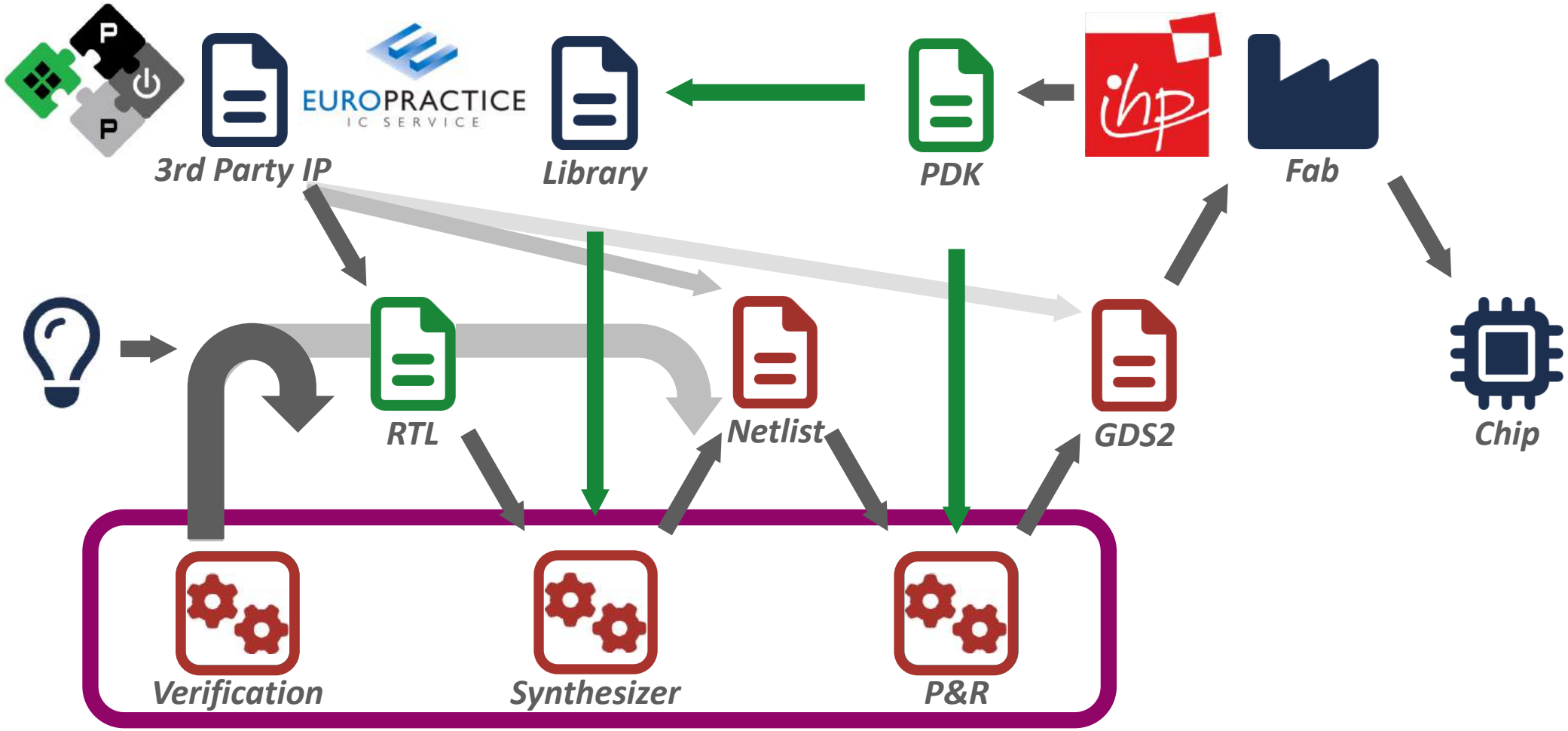
Fabs do not make PDK information accessible

Open PDKs are a key enabler for further development



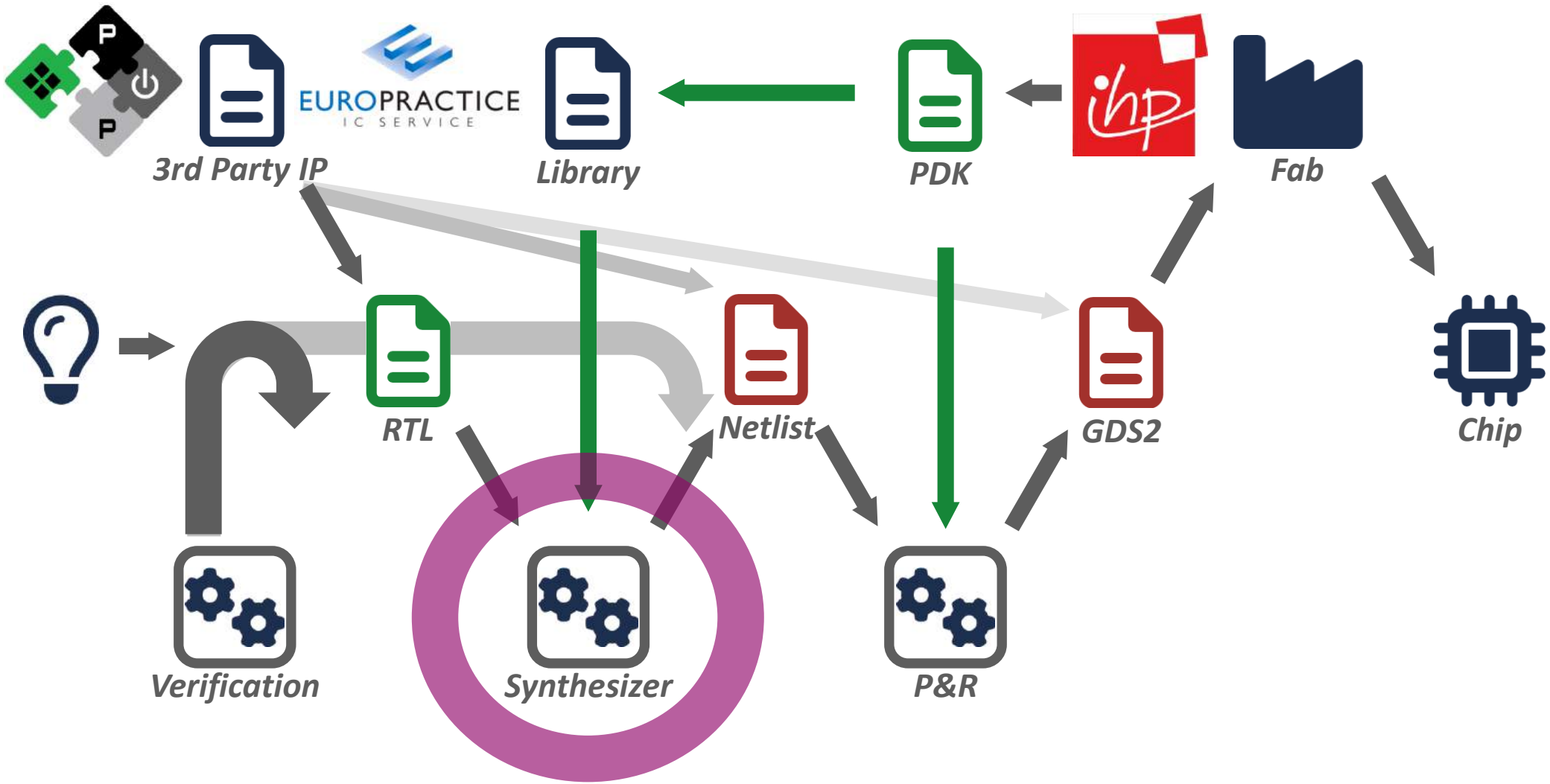
Icons taken from free icons from fontawesome.com

The output (and even scripts) of EDA vendors are closed



EDA vendors limit the output of their tools

Open-source community can develop EDA tools too!



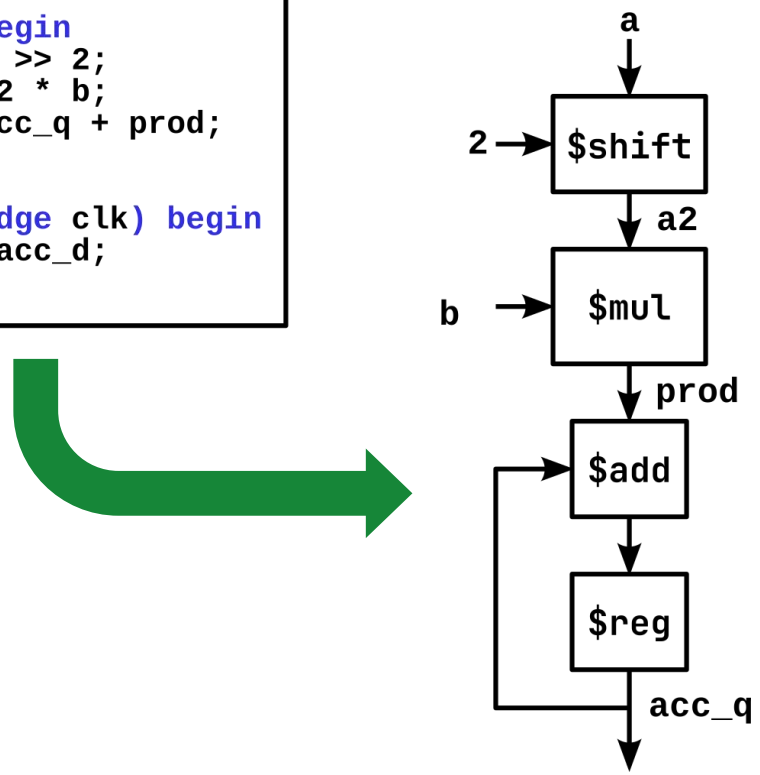
Icons taken from free icons from fontawesome.com

A look into the synthesis flow in Yosys

- **Elaboration**
  - Behavioral RTL to connected cells (structural)
- **High-level phase**
  - Cells are arithmetic operations
  - Fuse and transform operations

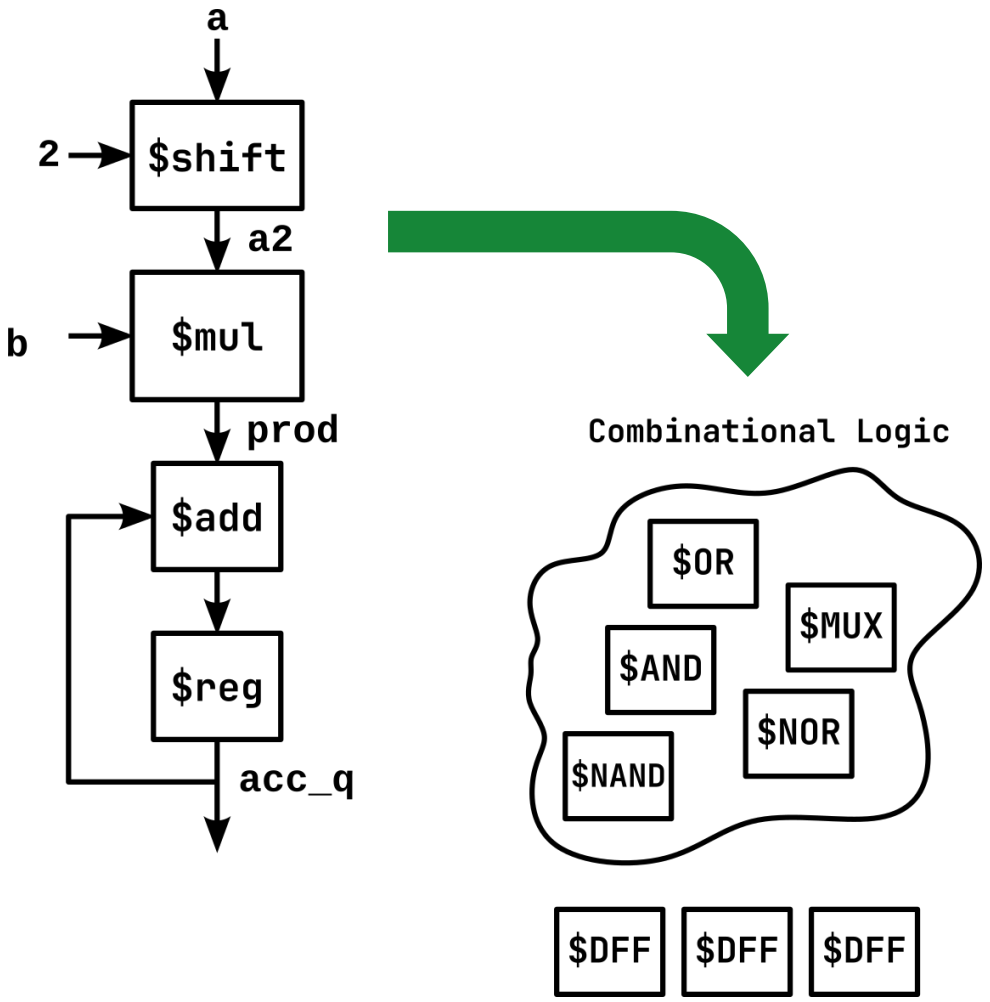
multipy-accumulate.v

```
wire [ 7:0] a, a2, b;
wire [15:0] prod;
reg  [15:0] acc_d, acc_q;

always @(*) begin
    a2    = a >> 2;
    prod  = a2 * b;
    acc_d = acc_q + prod;
end

always @(posedge clk) begin
    acc_q <= acc_d;
end
```



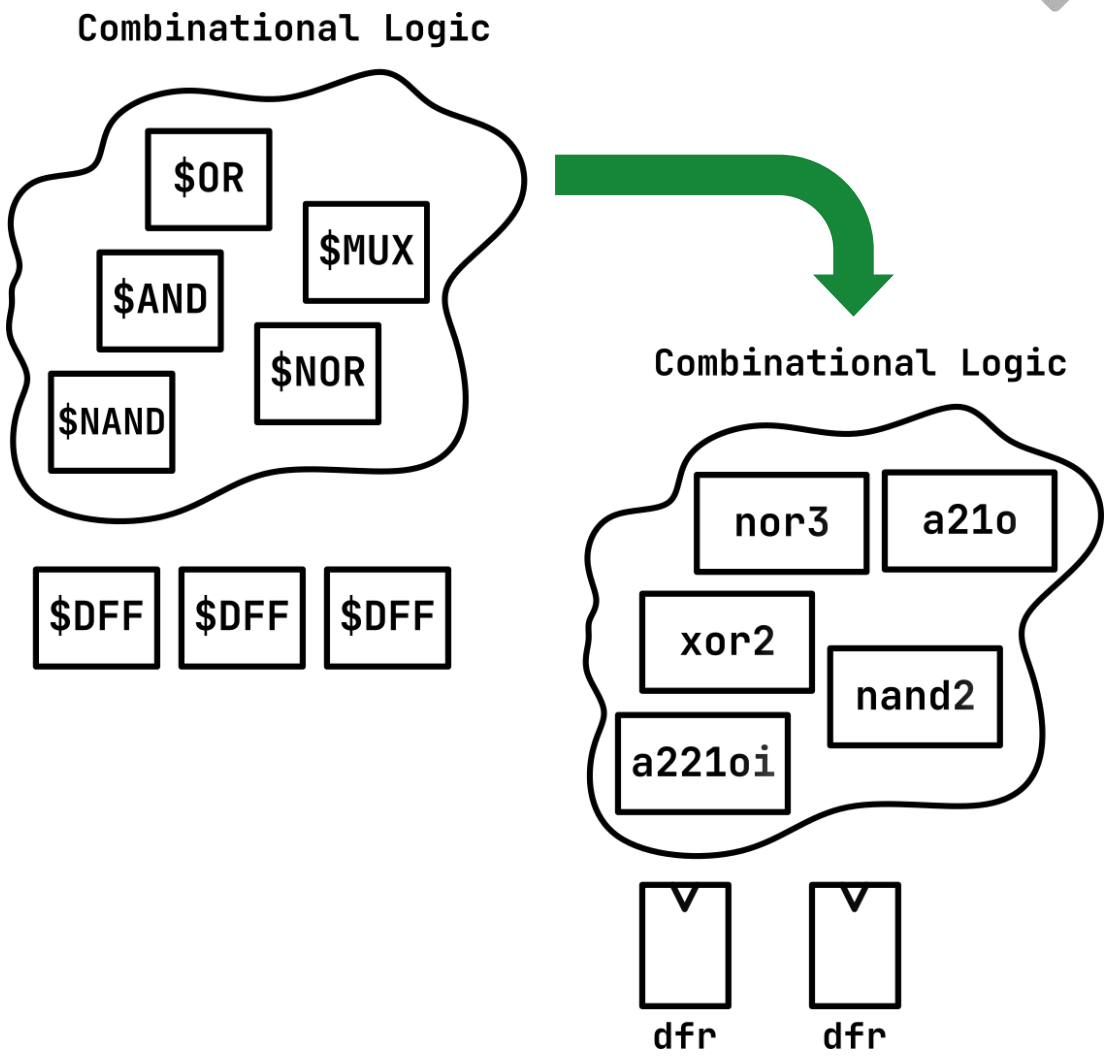
A look into the synthesis flow in Yosys

- **Elaboration**
  - Behavioral RTL to connected cells (structural)
- **High-level phase**
  - Cells are arithmetic operations
  - Fuse and transform operations
- **Generic gate phase**
  - Abstract standard cell library
  - Gate-level optimizations



A look into the synthesis flow in Yosys

- **Elaboration**
  - Behavioral RTL to connected cells (structural)
- **High-level phase**
  - Cells are arithmetic operations
  - Fuse and transform operations
- **Generic gate phase**
  - Abstract standard cell library
  - Gate-level optimizations
- **Technology mapping**
  - Performed in included tool called ABC
  - High-performance logic optimization
  - Mapping to standard cell library



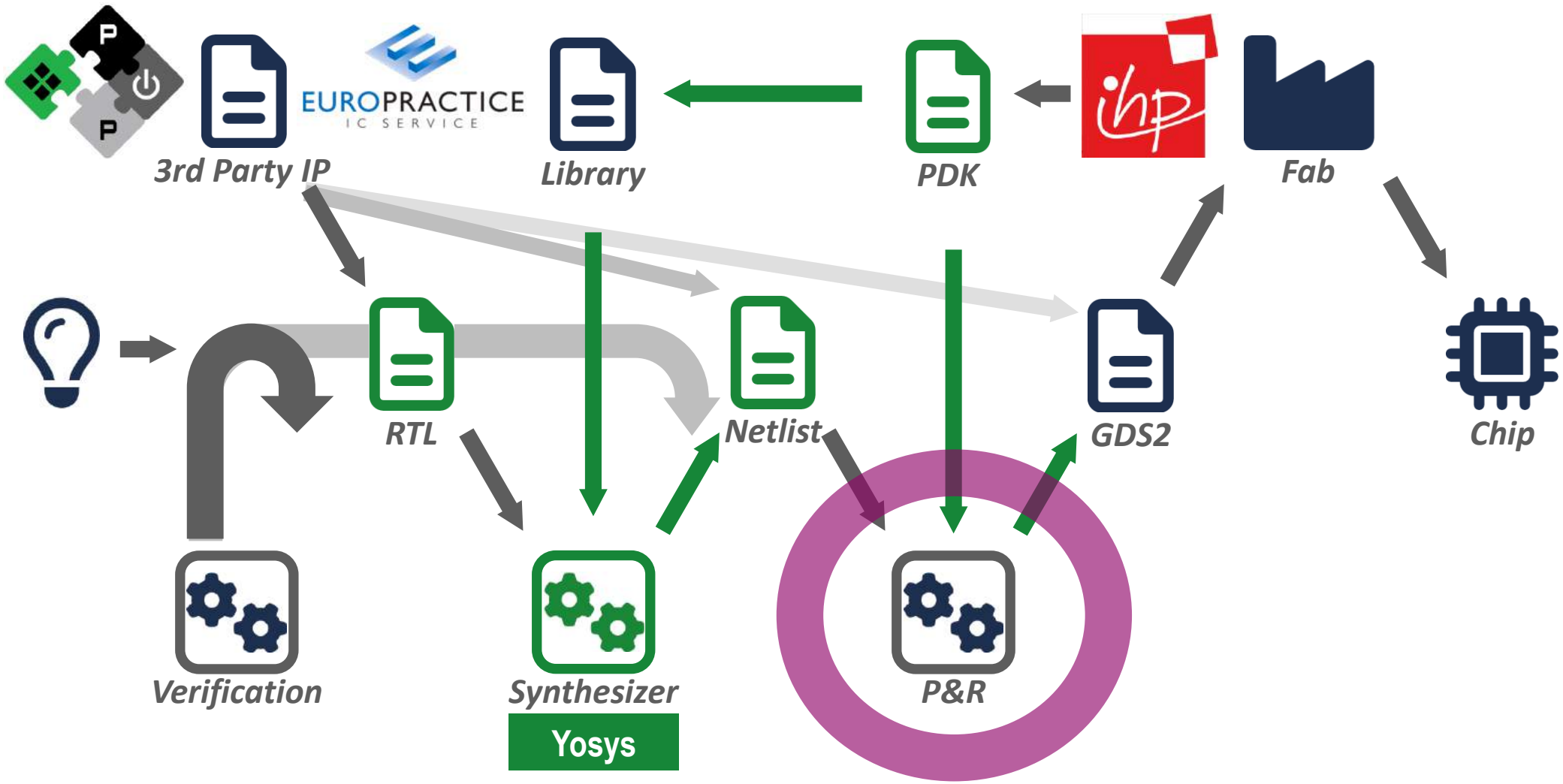
Yosys is structured, documented and maintained

- **Clear structure**
  - ‘Passes’ operate on current representation
  - Each pass is a file in a category (directory)
- **Guides for users and developers**
  - Starts with simple ‘how to use’
  - Ends with ‘how do I implement a pass’
- **Regular contributors**
  - YosysHQ employs developers
  - Other stakeholders also contribute often

YosysHQ

In <1week you can learn the basics and contribute meaningful improvements

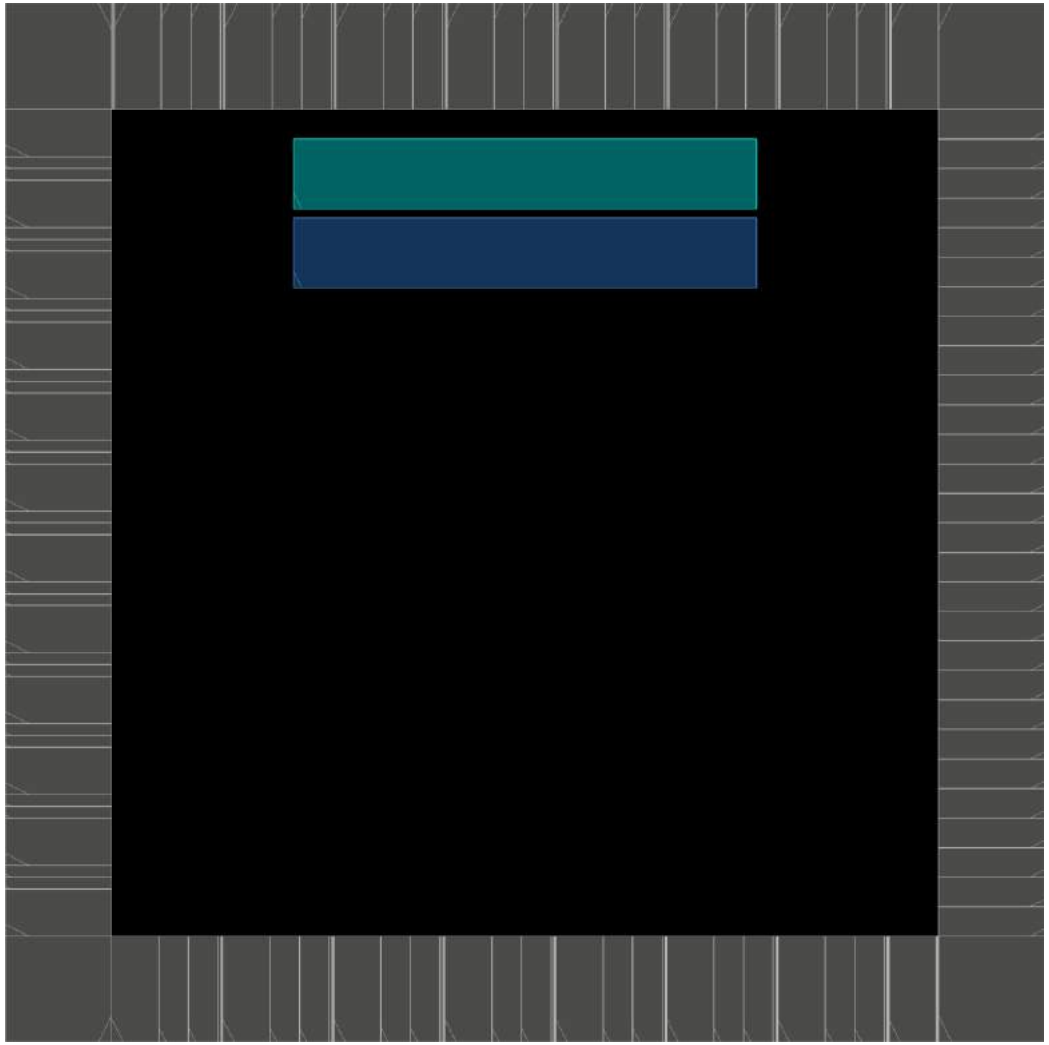
The output (and even scripts) of EDA vendors are closed



Icons taken from free icons from fontawesome.com

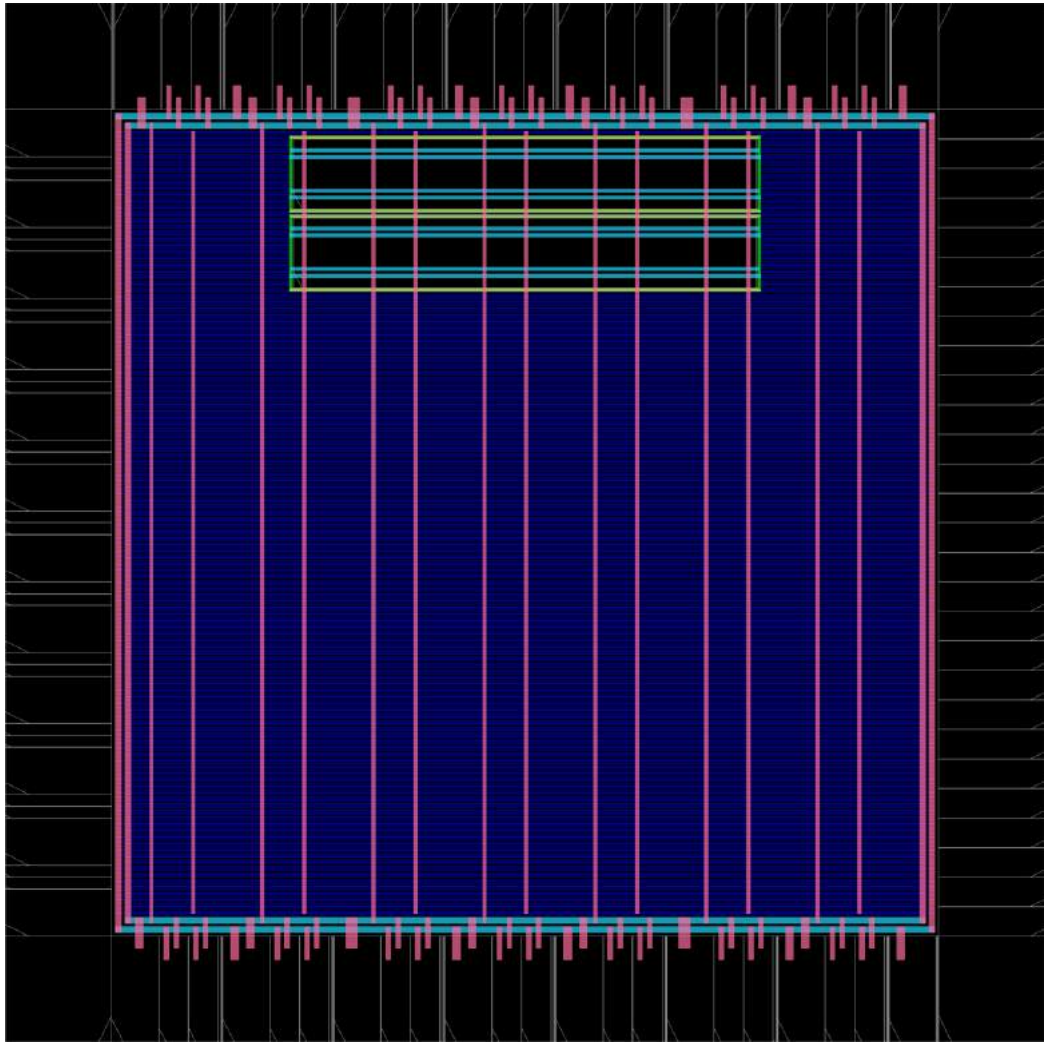
Place and Route flow in OpenROAD

1. **Floorplan**
   - Define size
   - Place pads and macros



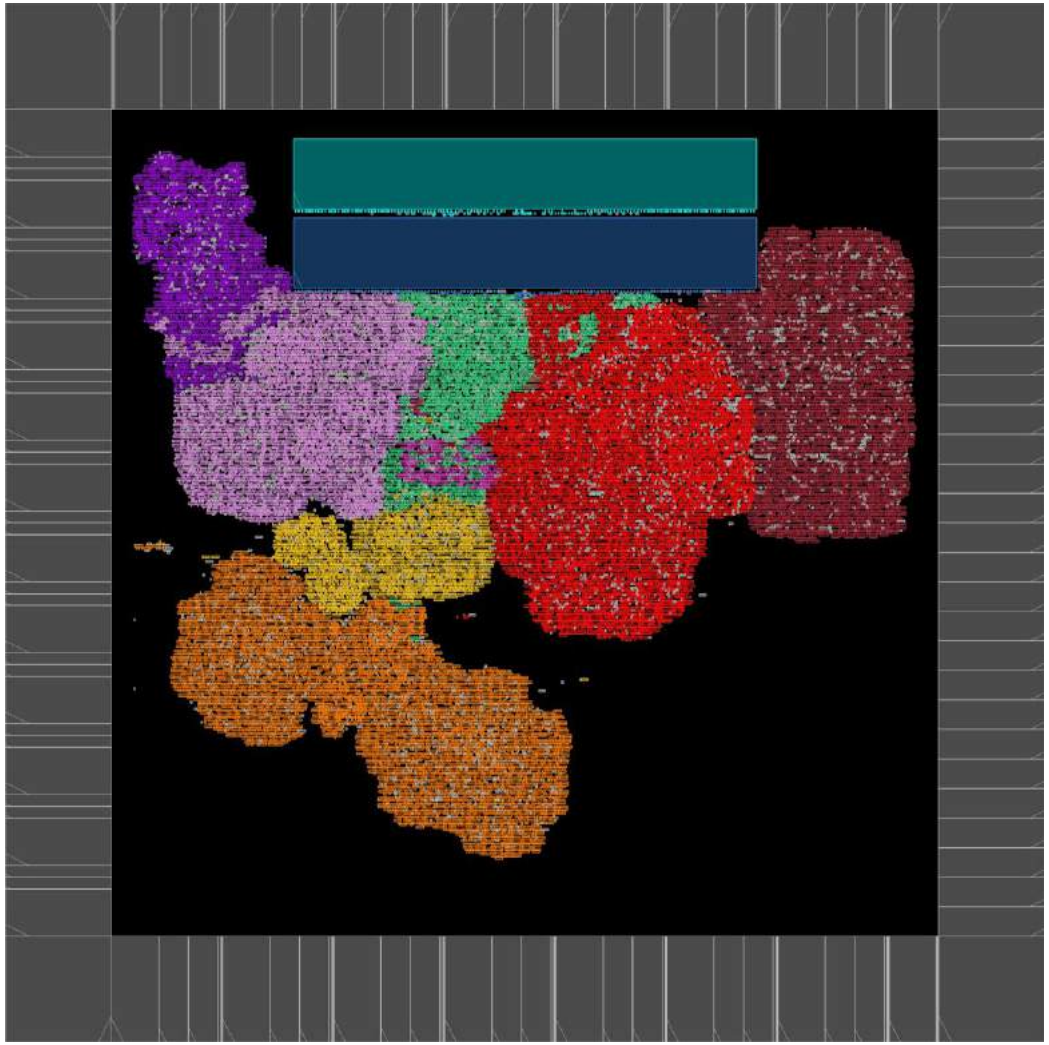
Place and Route flow in OpenROAD

1. **Floorplan**
   - Define size
   - Place pads and macros
2. **Power distribution**



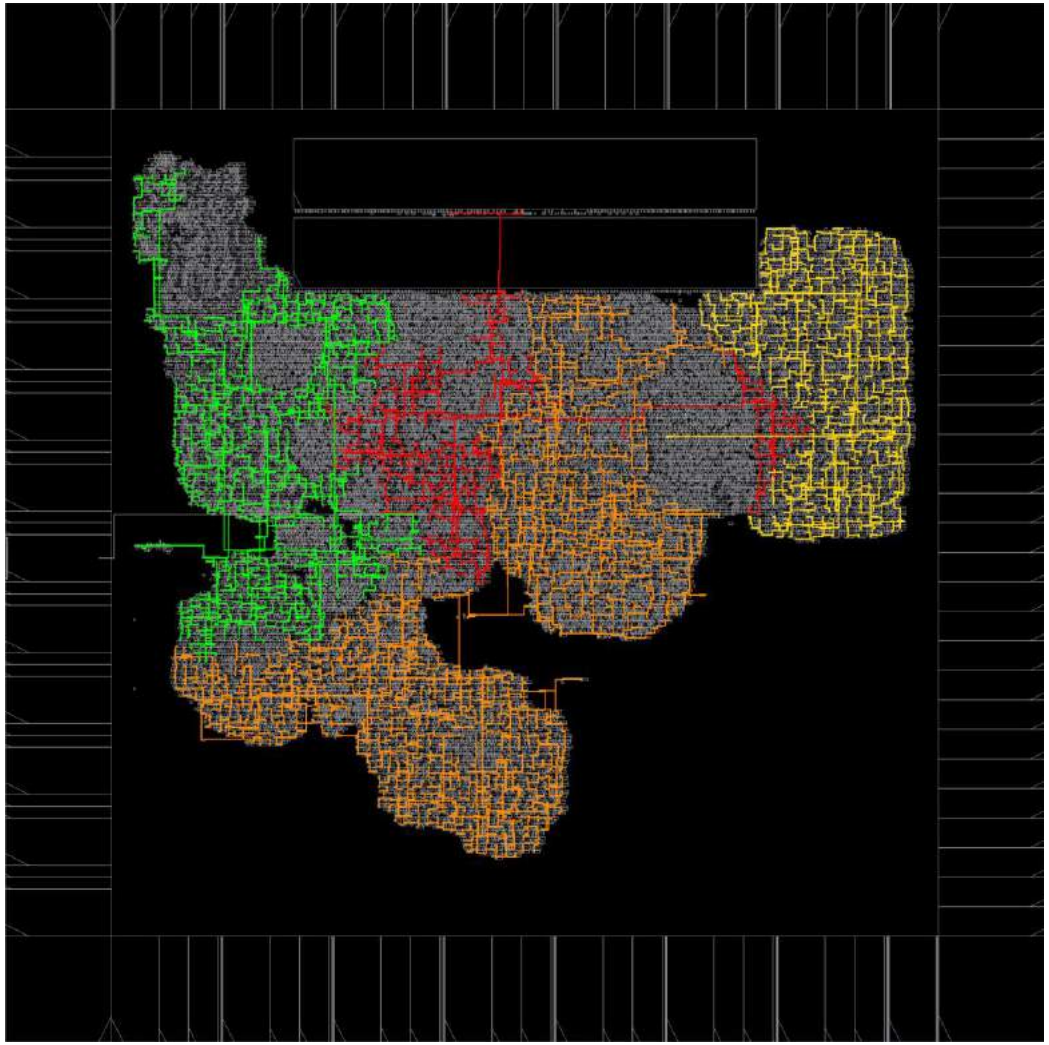
Place and Route flow in OpenROAD

1. **Floorplan**
   - Define size
   - Place pads and macros
2. **Power distribution**
3. **Placement**
   - Rough global placement
   - Legalize cell positions (detailed placement)



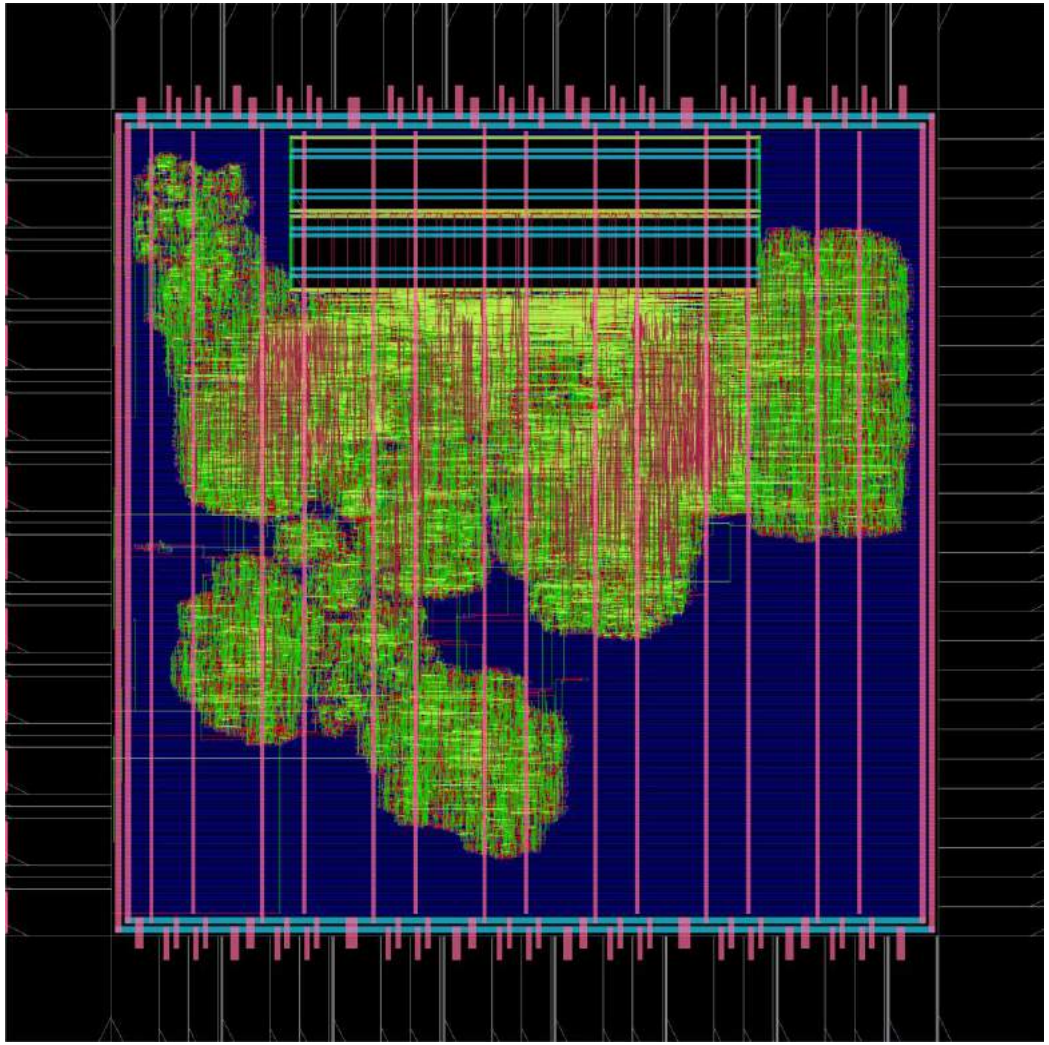
Place and Route flow in OpenROAD

1. **Floorplan**
   - Define size
   - Place pads and macros
2. **Power distribution**
3. **Placement**
   - Rough global placement
   - Legalize cell positions (detailed placement)
4. **Generate clock tree**



Place and Route flow in OpenROAD

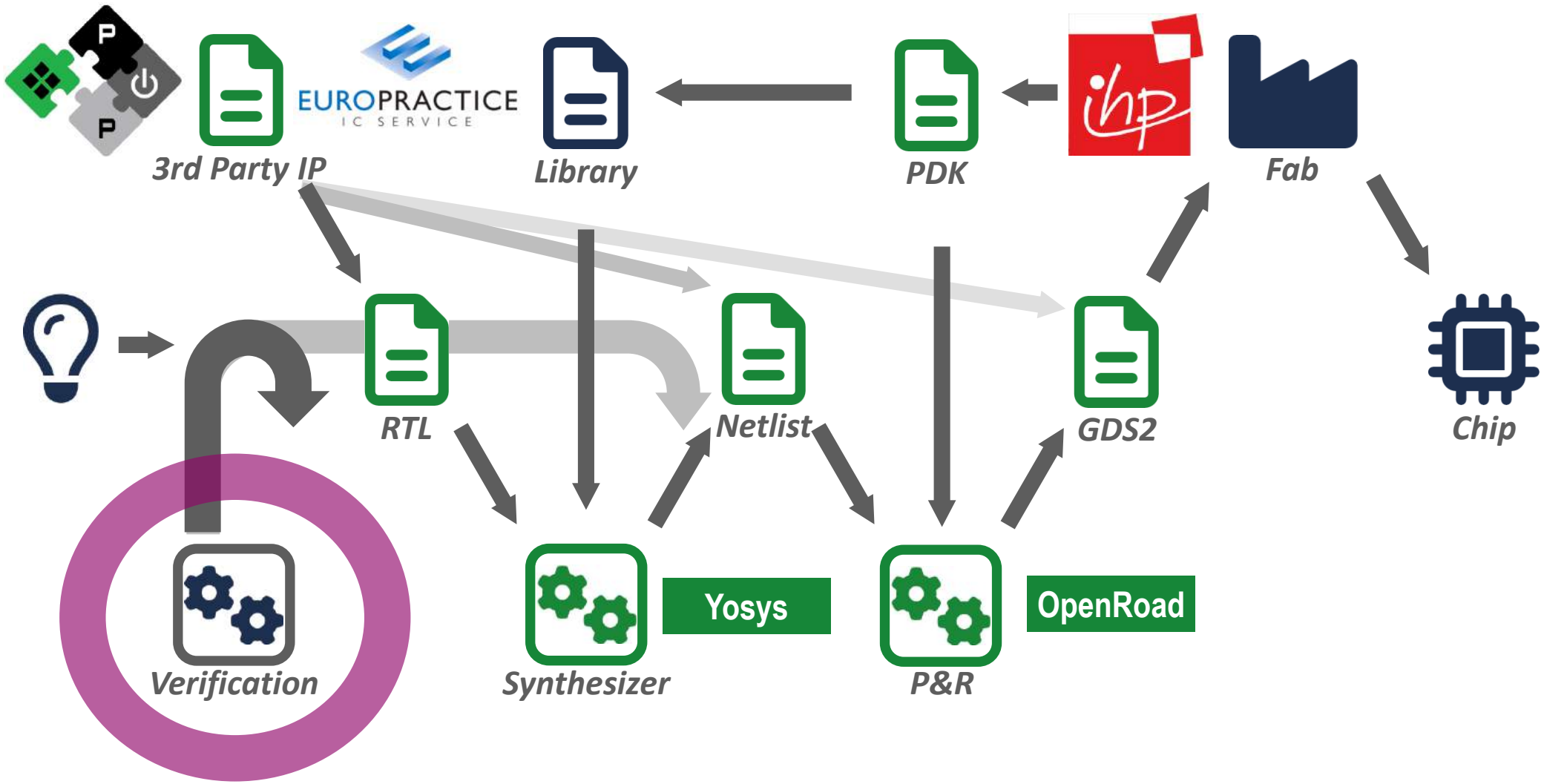
1. **Floorplan**
   - Define size
   - Place pads and macros
2. **Power distribution**
3. **Placement**
   - Rough global placement
   - Legalize cell positions (detailed placement)
4. **Generate clock tree**
5. **Routing**
   - Plan resource utilization for each wire
   - Create wires, fix violations (shorts etc)



OpenROAD: A Collection of Research Tools

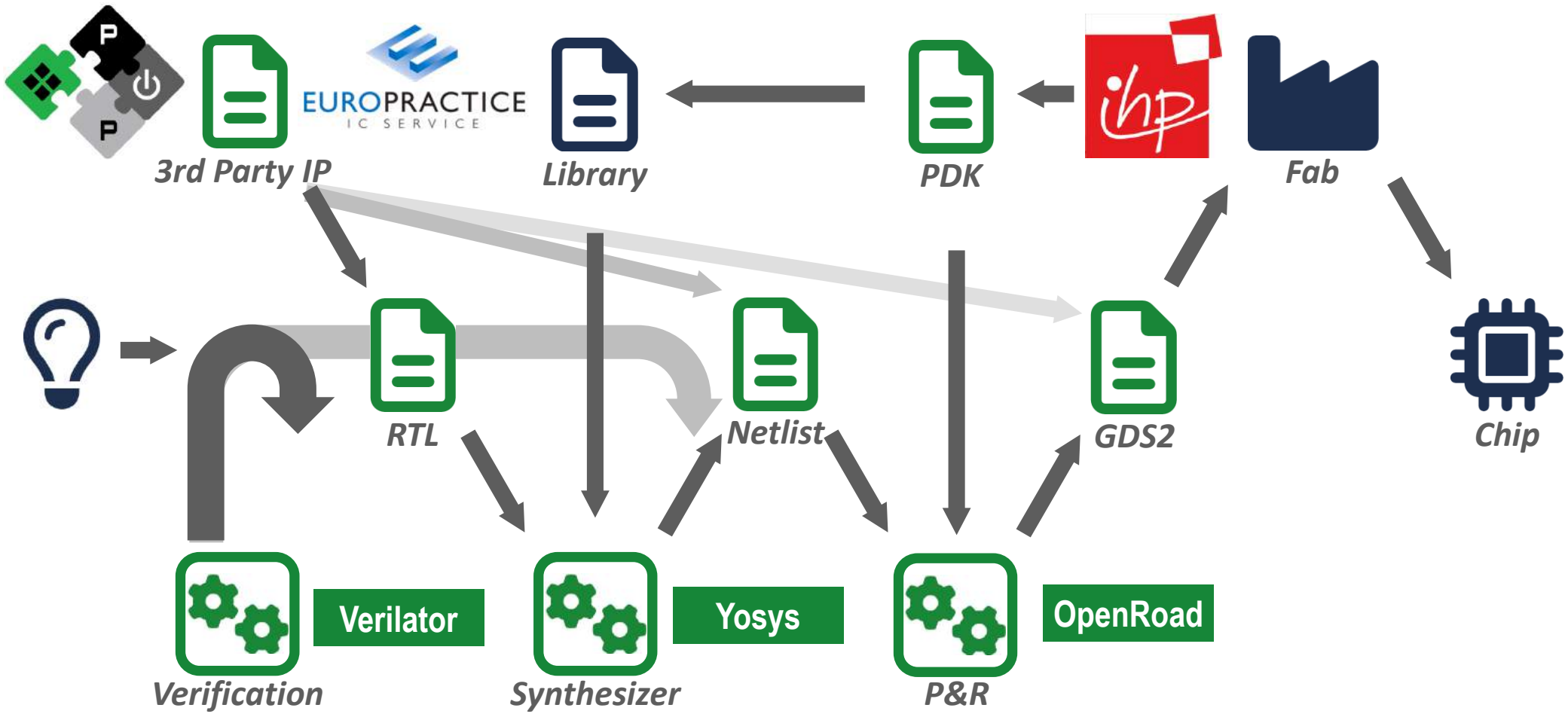
- **Research turned into a common flow**
  - Global place: RePlace
  - Global route: FastRoute
  - Clock tree: TritonCTS
- **Common openDB data structure**
  - Designed by industry professionals
  - Documented and tested
- **Supporting infrastructure around it**
  - CLI, GUI, reporting, metrics collection etc
  - Plugin system for easy extensibility

We need openness along the whole chain: RTL, EDA, PDK



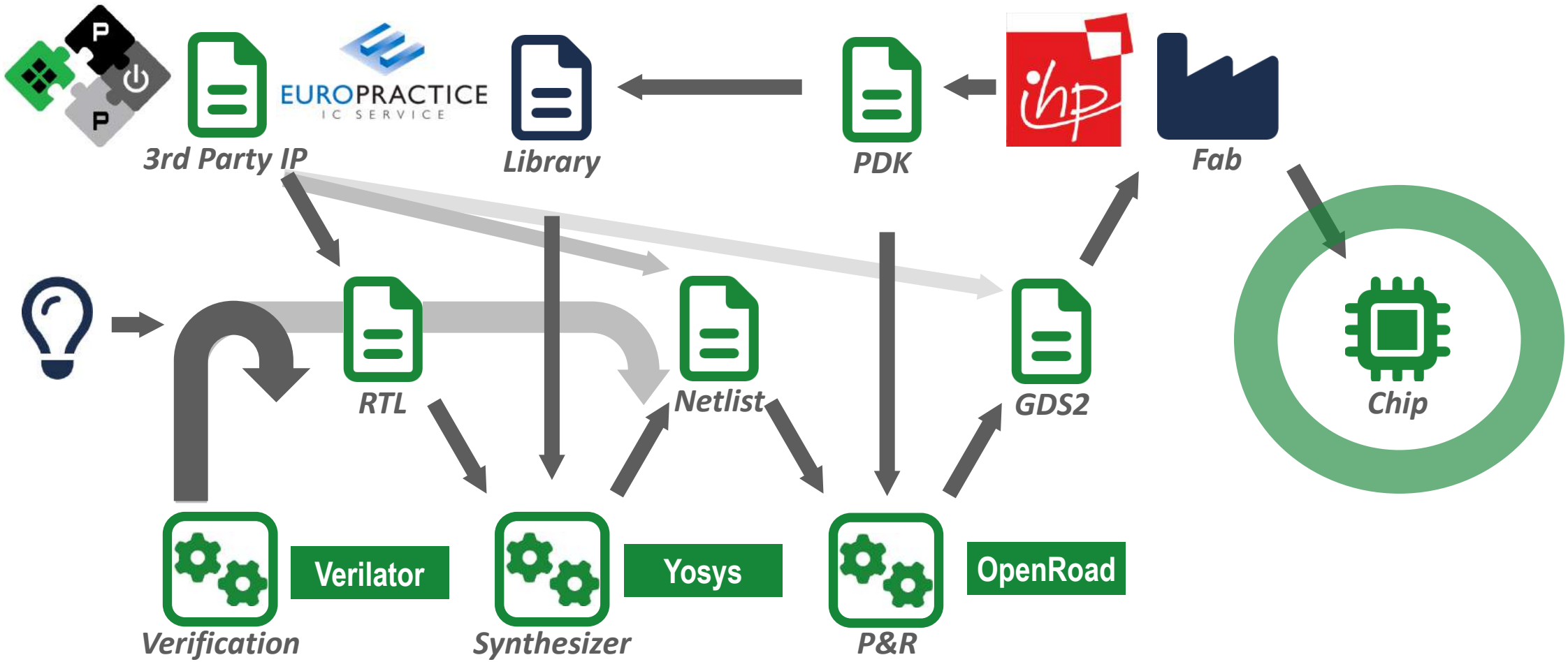
Icons taken from free icons from fontawesome.com

We need openness along the whole chain: RTL, EDA, PDK



Icons taken from free icons from fontawesome.com

We need openness along the whole chain: RTL, EDA, PDK



We are getting there, first fully open chips are underway

Icons taken from free icons from fontawesome.com

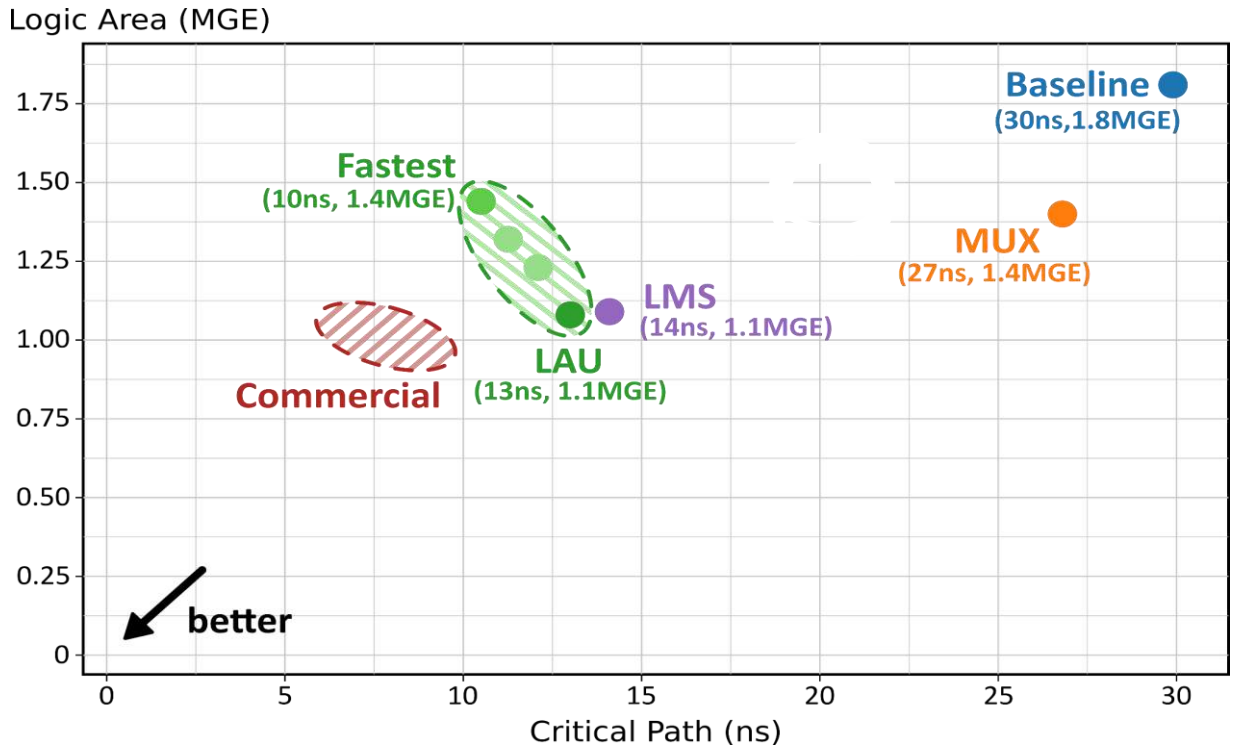
Meet Basilisk: Open **RTL**, Open **EDA**, Open **PDK**

- **Designed in IHP 130nm OpenPDK**
  - 6.25mm x 5.50mm
  - 60MHz
    - 1.08 MGE logic, 60% density
    - 24 SRAM macros (114 KiB)
- **CVA6 based SoC**
  - Runs and boots Linux
- **Active collaboration with**

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

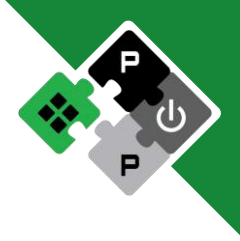
Working with open-source EDA groups to close the gap!

- **Basilisk is the first end-to-end open-source Linux-capable RV64 SoC**
  - DRAM interface & rich IO (USB 1.1, VGA, SPI, …)
  - Silicon-proven, configurable, MGE-scale design
- **Improved FOSS EDA flow**
  - SV-to-Verilog chain @ **<2min** runtime
  - Yosys synthesis:
    → **1.1 MGE** (**1.6×**) @ **77 MHz** (**2.3×**)
    → **2.5×** less runtime, **2.9×** less peak RAM
  - OpenROAD P&R: tuning
    → **-12%** die area, **+10%** core utilization



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

Benefits of end to end openness



Research

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

Industry

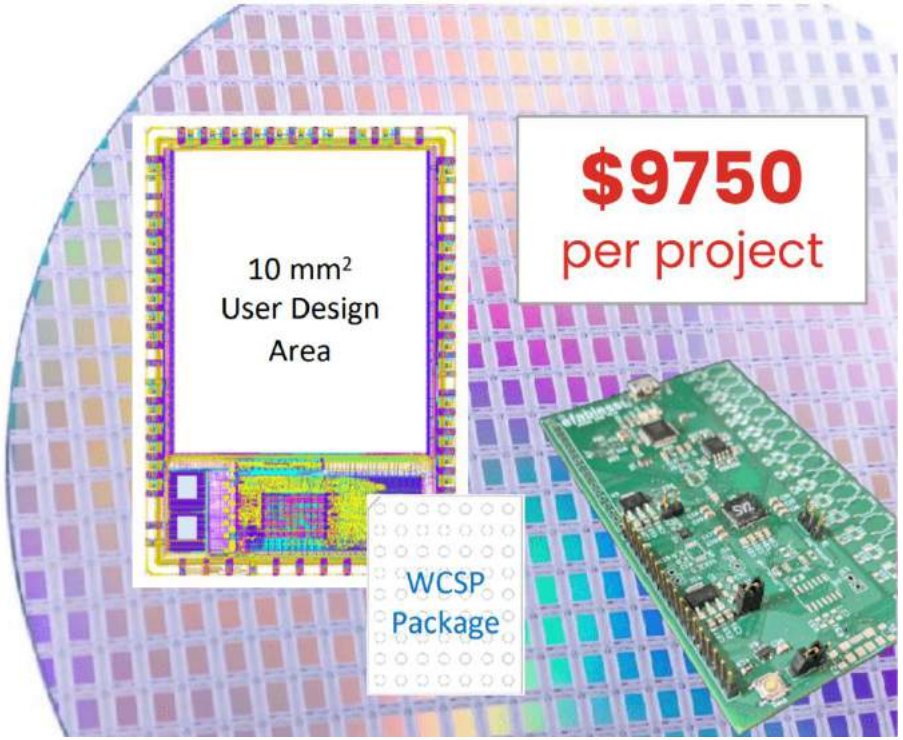
- Transparent chain of trust, sovereignty
- Lower initial cost
- **Faster research → product**

Education

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

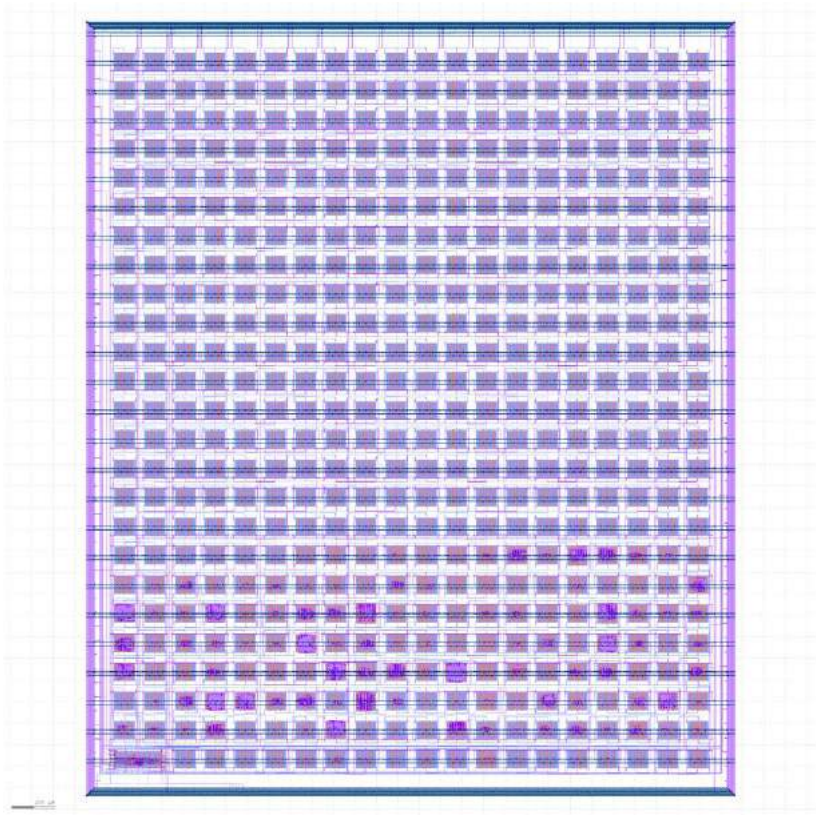
Education is gaining momentum

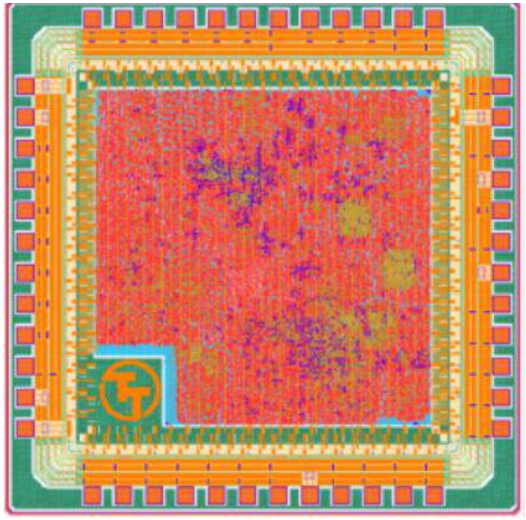
chipignite



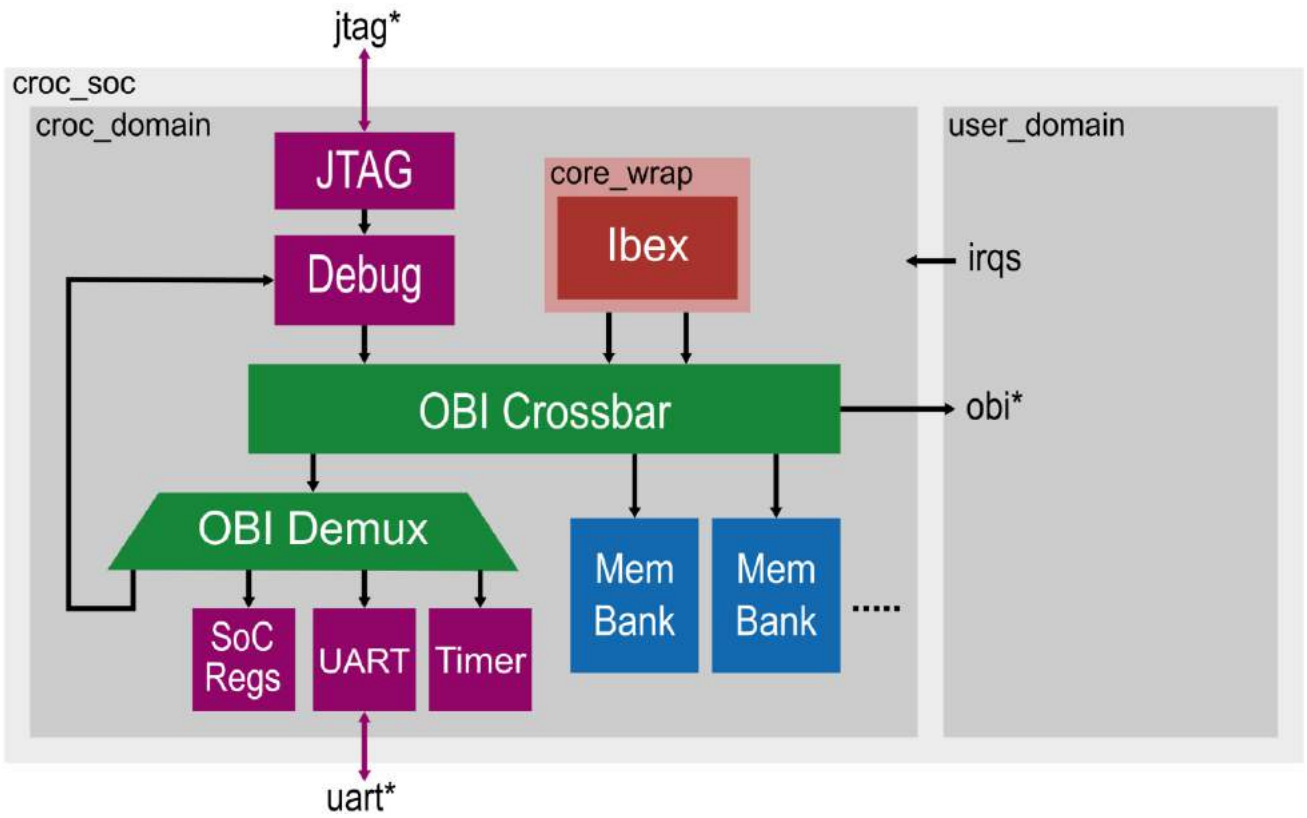
Tiny Tapeout

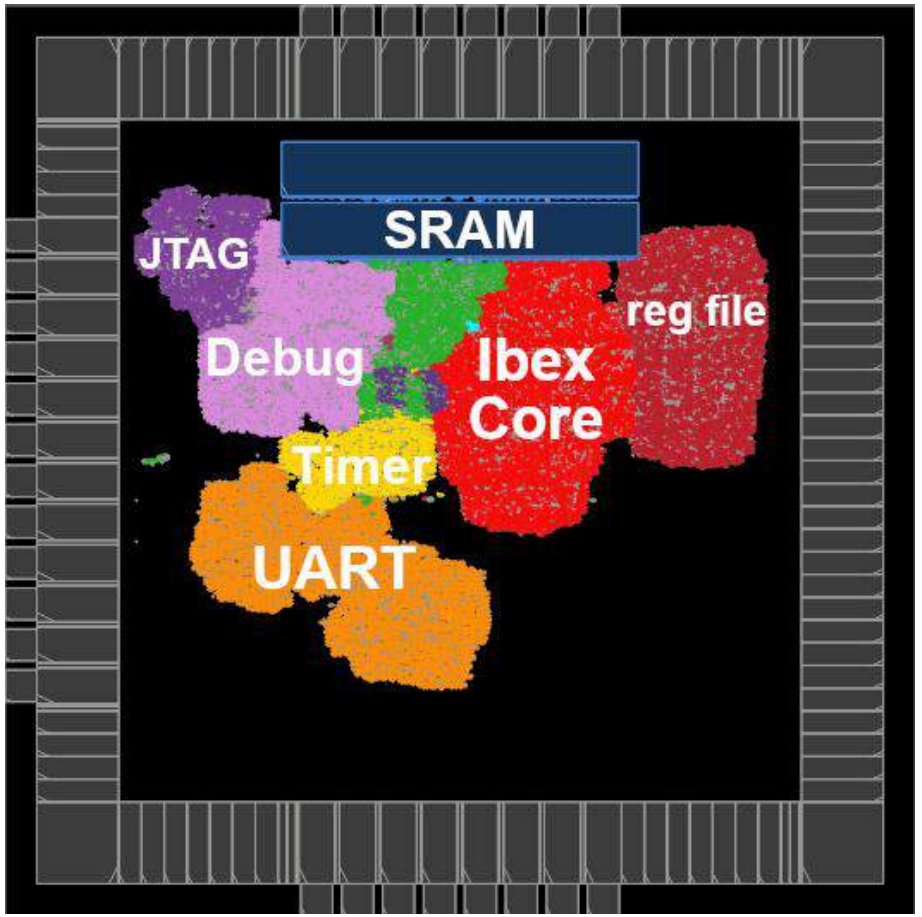
$300 (currently $150)





Croc SoC: A simple chip for students





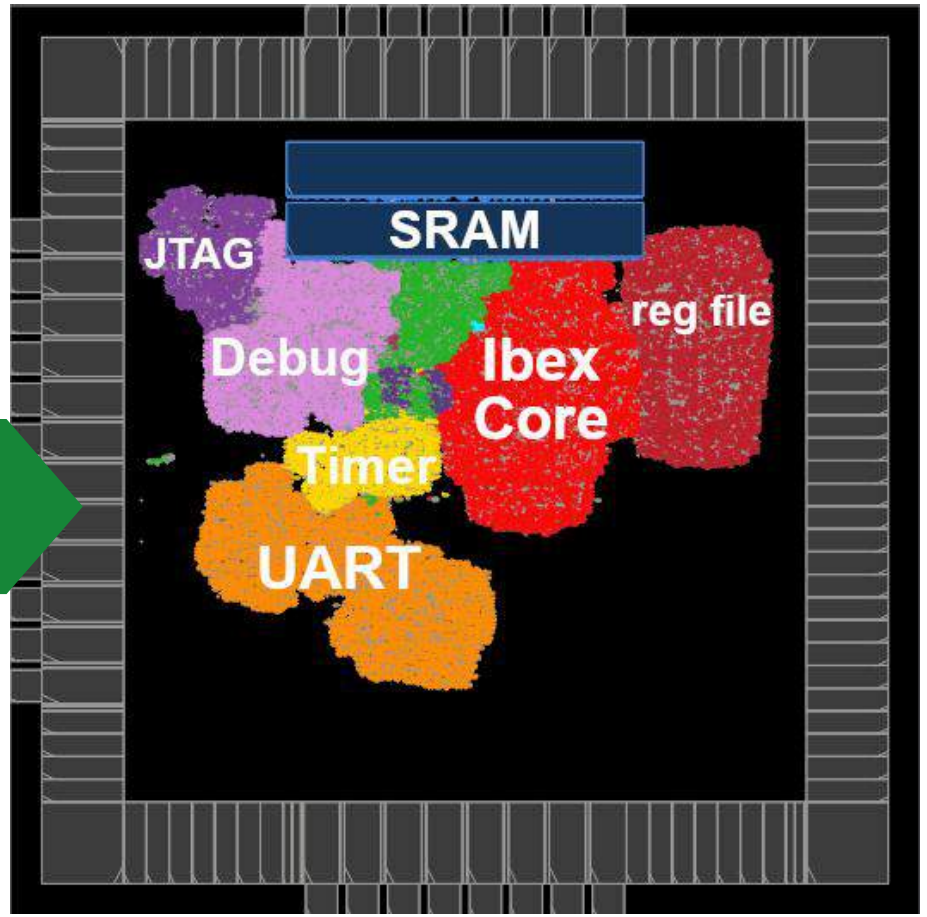
github.com/pulp-platform/croc

Croc SoC: A simple chip for students

- **Croc is simple to understand**
  - Everything in one repository
  - Plain SystemVerilog
  - (soon) guides from students for students
- **Croc is flexible**

Used for ETH Zürich VLSI 2 lecture starting 2025

- **Croc flow is easy to run**
  - Runs on older laptops (<4GB RAM)
  - Tools in a docker container
    Works on Linux, Windows and MacOS
  - 4 Make commands from RTL to GDS



`github.com/pulp-platform/croc`

Final words

- **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