

# Basilisk: A 34 mm<sup>2</sup> End-to-End Open-Source 64-bit Linux-Capable RISC-V SoC in 130nm BiCMOS

Integrated Systems Laboratory (ETH Zürich)

Authors:

**Philippe Sauter**<sup>1</sup> [phsauter@iis.ee.ethz.ch](mailto:phsauter@iis.ee.ethz.ch)

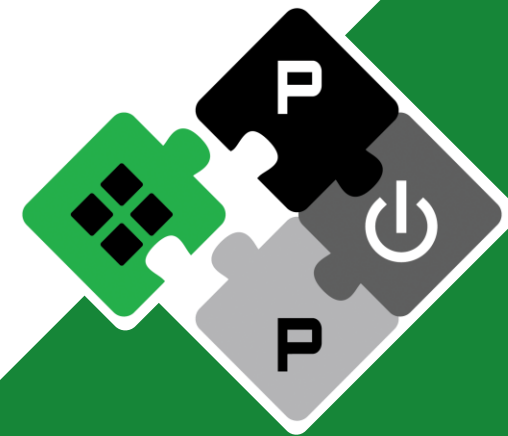
**Thomas Benz**<sup>1</sup>, **Paul Scheffler**<sup>1</sup>, **Martin Poviser**,  
**Frank K. Gürkaynak**<sup>1</sup>, **Luca Benini**<sup>1,2</sup>

<sup>1</sup>ETH Zurich

<sup>2</sup>University of Bologna

**PULP Platform**

Open Source Hardware, the way it should be!





## Open-source as enabler

- **Academia:** NDA-free collaboration on designs and tools
- **Education:** Widely accessible hands-on chip design
- **Industry:** Zero-trust verification, license-free deployment

## Explore feasibility for larger Linux-capable SoCs

- **Previously:** OS EDA tools used for tapeout of small designs
- **Now:** 4.8× larger than largest previously published design
- **Novelty:** Significant improvements in QoR and performance of open-source EDA tools and flow



# Main Points



- First end-to-end permissive, free and open-source **Linux-capable SoC** with an **application-class core** and **rich peripherals**
- Newly developed **Yosys-Slang** enables synthesis of complex, **industry-grade SystemVerilog RTL**
- **Reproducible** and **sharable** high-quality designs for collaboration and research



[github.com/pulp-platform/cheshire-ihp130-o](https://github.com/pulp-platform/cheshire-ihp130-o)



# Basilisk is Built from Open-Source IPs

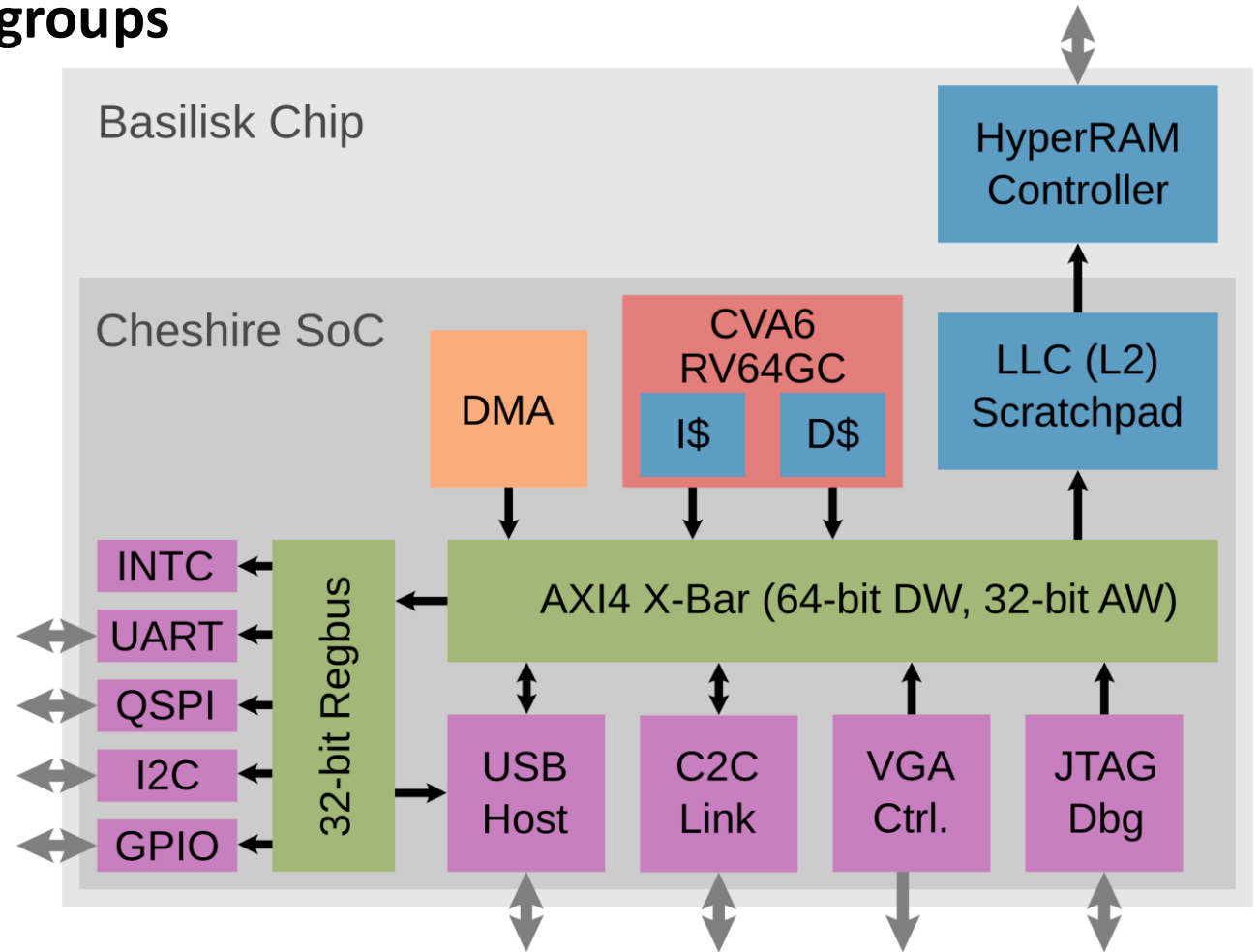


- **Permissive licensed RTL from various groups**

- PULP Platform (Cheshire, AXI, Hyperbus...)
- OpenTitan (SPI, I2C)
- OpenHW Group (CVA6 core)
- SpinalHDL (USB)

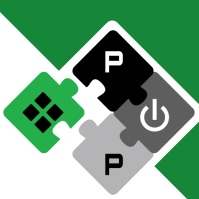
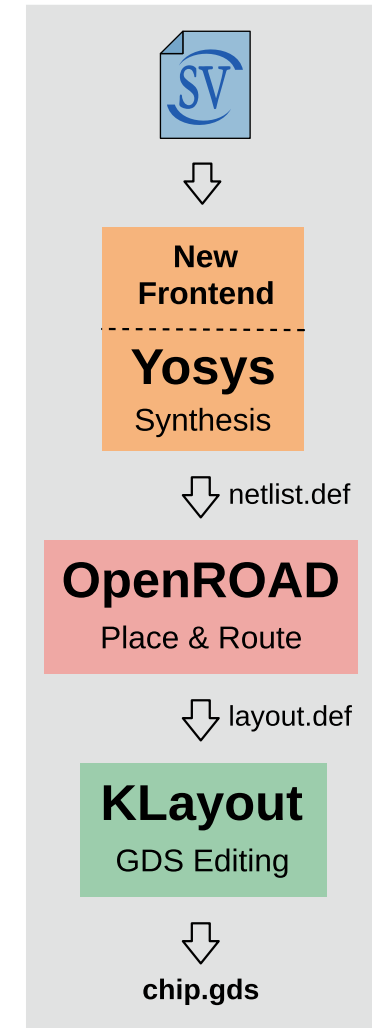
- **Architecture**

- RV64GC-compliant CVA6 core
- Cheshire SoC provides vital peripherals
- 4-way 16KiB L1-D and L1-I cache
- 4-way 64KiB LLC / Scratchpad memory
- Hyperbus DRAM controller achieving transfer speeds up to 124MB/s



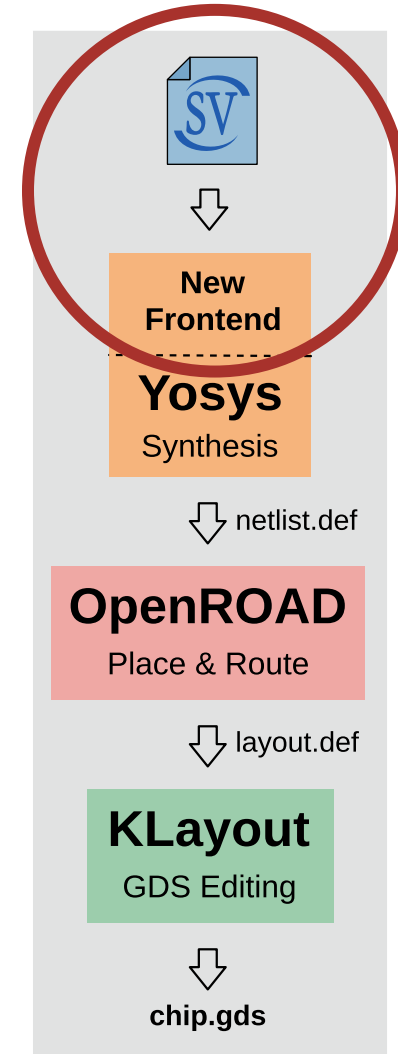
# Five Key Tools in our Open-Source Flow

- **Yosys-Slang:** Our newly developed SystemVerilog frontend based on Slang, parses and elaborates industry-grade RTL
- **Yosys:** Synthesizer integrates logic optimization and mapping of high-level operators (eg multiply) to specific implementation
- **ABC:** Berkeleys logic optimization tool works on And-Inverter Graphs (AIG) and maps to the technology's standard cells
- **OpenROAD:** Floorplanning, cell placement and routing
- **KLayout:** DRC and LVS checks and final processing of layout



# Yosys-Slang: Our New SystemVerilog Frontend

- **Based on Slang: the SystemVerilog language service**
  - Slang is an established SystemVerilog library
  - It demonstrates near complete coverage of the SystemVerilog spec
  - Development is driven directly from the language standard
  - Slang makes it easy to build SystemVerilog tools
- **Yosys-Slang is focused on quality at all levels**
  - Written to take full advantage of Slang's performance
  - Unit-tests and full-design tests to check correctness of elaboration
  - Elaboration decisions made to improve quality of results for synthesis

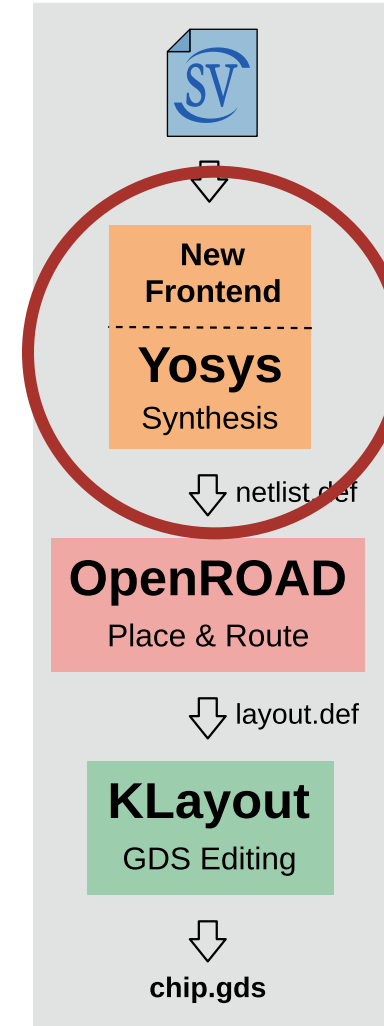


[github.com/povik/yosys-slang](https://github.com/povik/yosys-slang) 



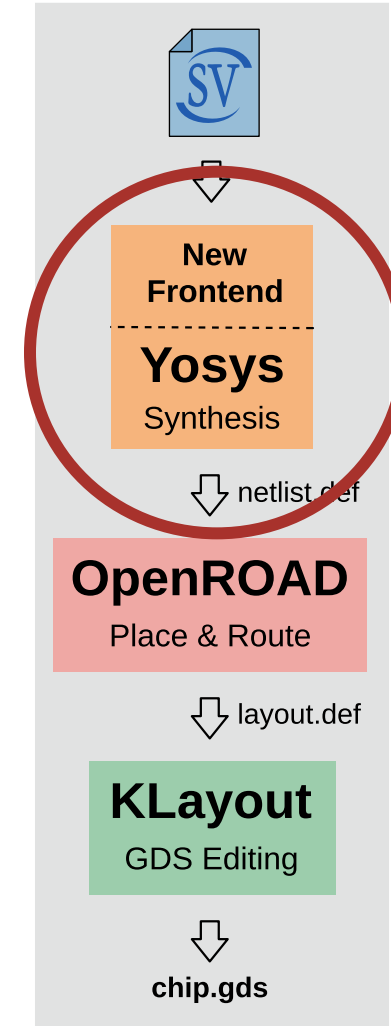
# Yosys Synthesis Improvements

- **Lazy man's synthesis for high effort optimization**
  - We implement a new ABC flow leveraging high-effort synthesis techniques
  - Logic optimization technique developed by Yang et al.
  - Available in ABC logic optimizer integrated into Yosys
  - Idea: precompute near-optimal structures for given functions, during synthesis select best one and map to found structure
- **Optimize handling of bit-select operator**
  - We analyze Yosys' output to find shortcomings
  - We improved mapping of bit-select operator to multiplexers trees reducing memory usage and runtime while also reducing logic cell area



# Results from Yosys Synthesis Improvements

- Compared to the tools open-source reference flow
  - **2.3×** reduction in critical path length
  - **1.6×** reduction in logic cell area after synthesis
  - **2.5×** runtime reduction
- **Competitive results to previous commercial implementations**
  - Previous Cheshire SoC implemented with commercial tools
  - They report a technology-independent critical path of 46 logic levels
  - We achieve a **critical path of 51 logic levels**



# OpenROAD Backend Flow Tuned for Basilisk

- **-12% die area vs open-source reference flow**
  - Based on the OpenROAD-flow-scripts reference flow
  - The Flow and used hyperparameters are tuned for our design
- **Tuned hyperparameters in OpenROAD**
  - Strength of routing driven placement
  - Strength of timing driven placement
  - Per-layer routing resource reduction
  - Per-module density reduction via cell padding
  - And more...



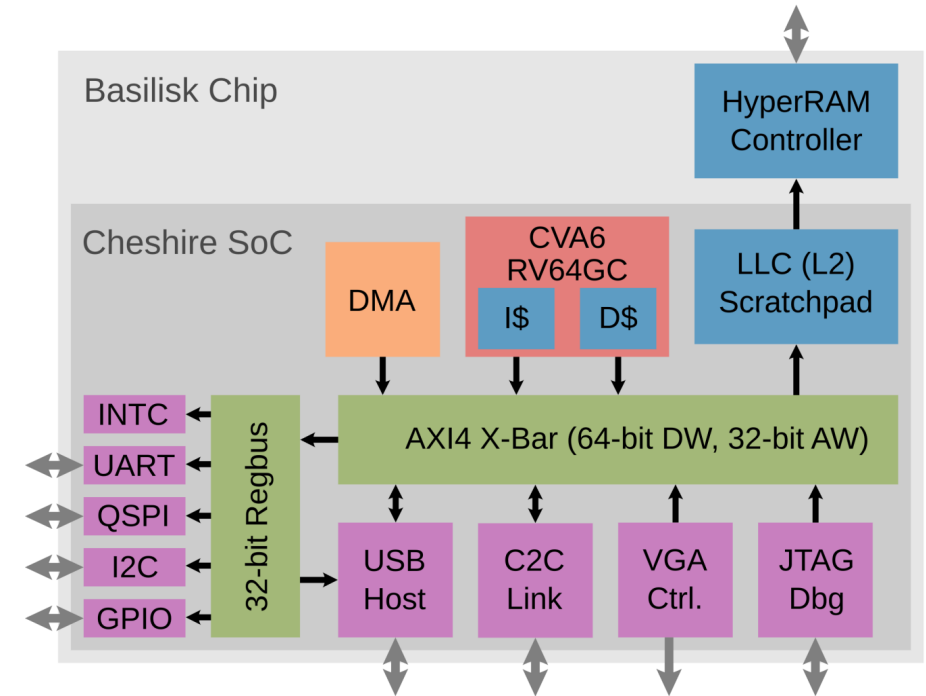
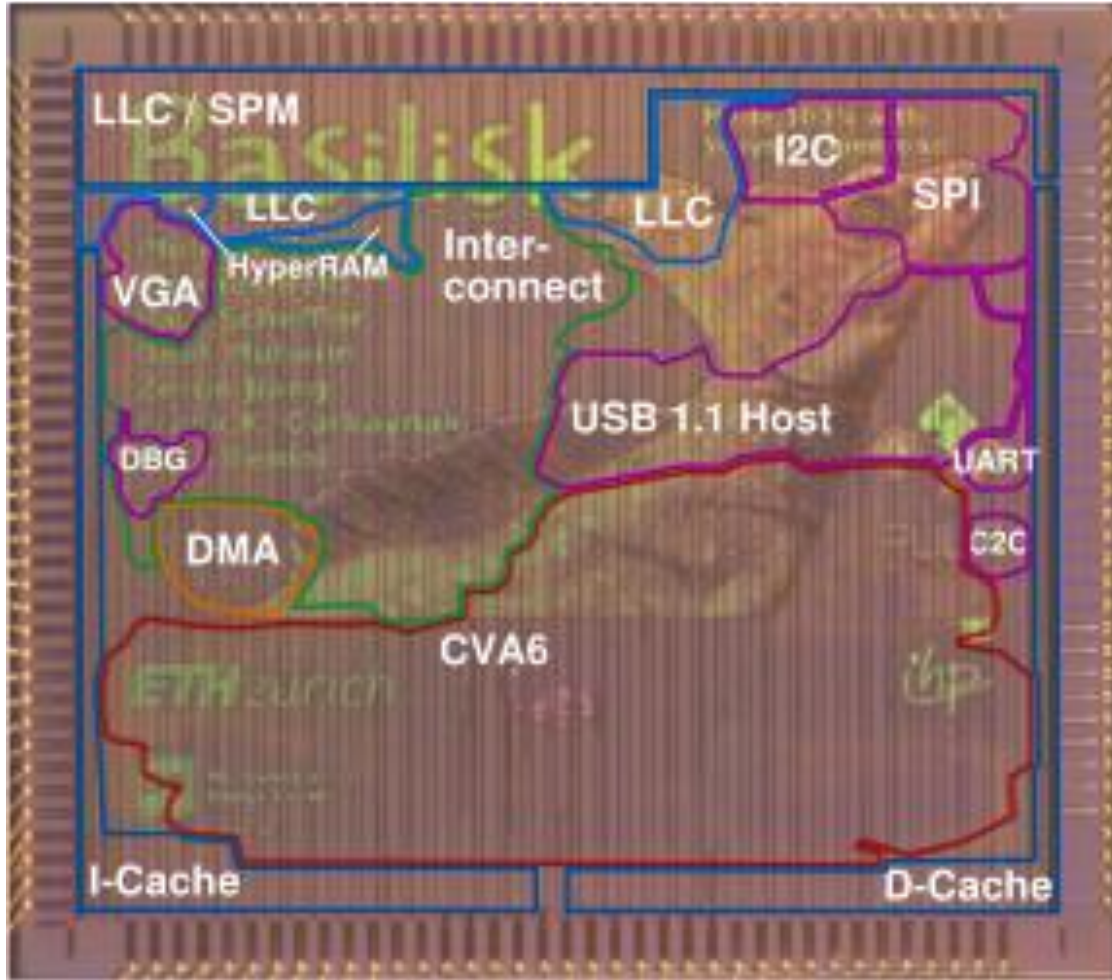
# Basilisk Die Shot



Come see the die in person at our HotChips poster!



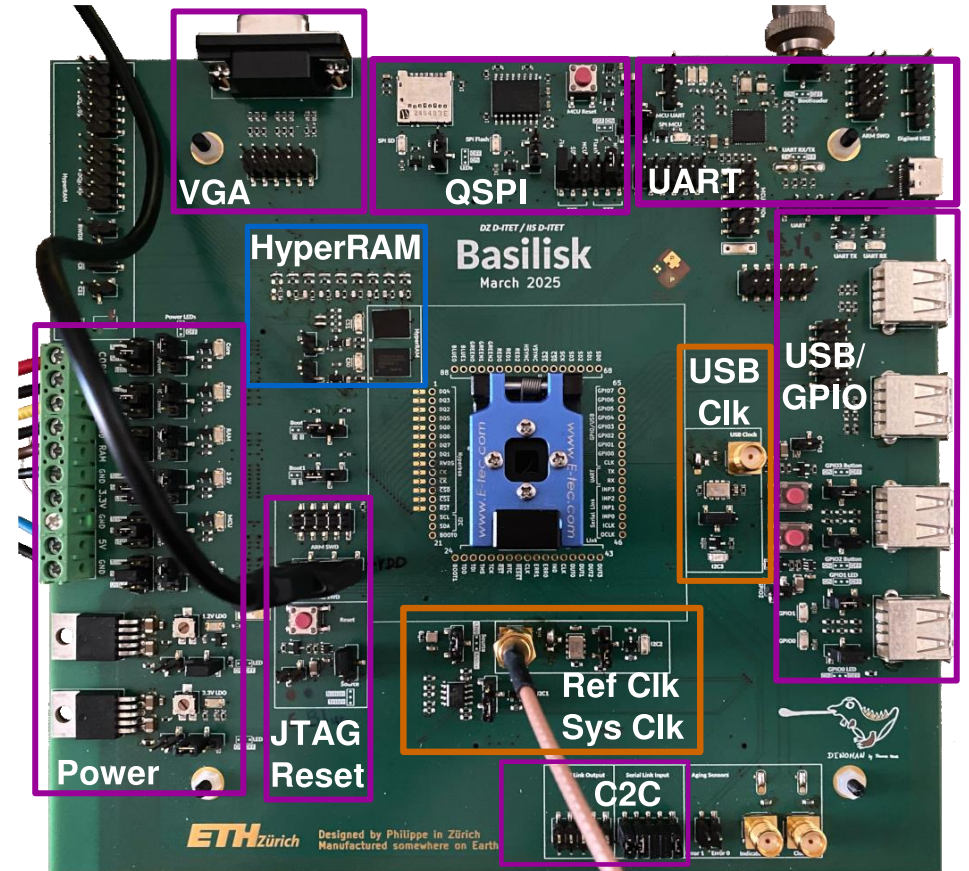
# Basilisk Die Shot: Annotated



# Silicon Testing on Demo Board



- **Basilisk demo board to fully test our chip**
  - Designed in the open-source PCB EDA tool KiCAD
  - Chip selects between multiple different boot sources
  - On-board configurable clock sources
  - All peripherals (VGA, USB, UART...) work in a Linux environment running on our chip
- **Open-Source testing orchestration**
  - Stimul application and measurements are orchestrated and timed by DUTCTL
  - DUTCTL is an open-source measurement framework



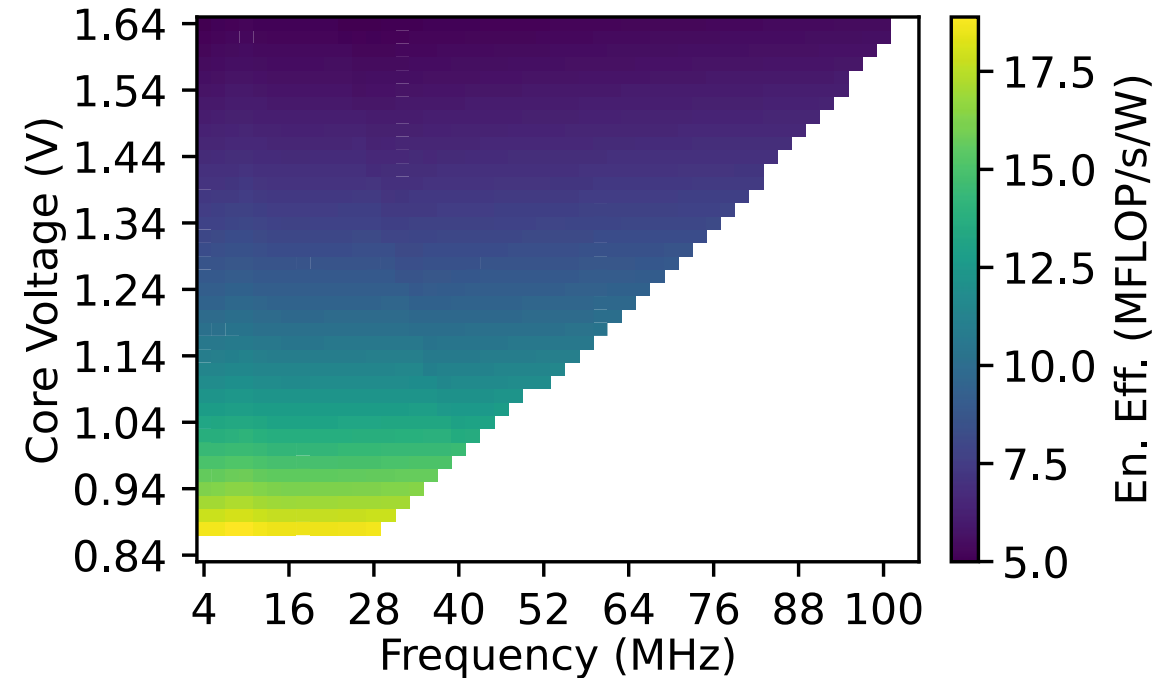
Come by our poster for a live demo!



# Measurement Results



- **Test workload: 48x48 FP64 GEMM**
  - Measurements from 10 packaged Basilisk chips
- **64 MHz operating frequency**
  - At 1.2V nominal voltage, room temperature
  - Matching final timing analysis in OpenROAD
- **102 MHz peak operating frequency**
  - At 1.64V max voltage, room temperature
- **Peak efficiency of 18.9 MFLOP/s/W**
  - Achieved at 0.88V core voltage



# Conclusion



- First end-to-end permissive, free and open-source **Linux-capable SoC** with an **application-class core** and **rich peripherals**
- Newly developed **Yosys-Slang** enables synthesis of complex, **industry-grade SystemVerilog** RTL
- **Reproducible** and **sharable** high-quality designs for collaboration and research



More resources online

