

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

Philippe Sauter<sup>✉\*</sup>, Thomas Benz<sup>✉\*</sup>, Paul Scheffler<sup>✉\*</sup>, Martin Povišer<sup>✉</sup>, Frank K. Gürkaynak<sup>✉\*</sup>, Luca Benini<sup>✉†</sup>  
<sup>\*</sup> ETH Zurich, Switzerland <sup>†</sup> University of Bologna, Italy

**Abstract**—End-to-end open-source electronic design automation (OSEDA) enables a collaborative approach to chip design conducive to supply chain diversification and zero-trust step-by-step design verification. However, existing end-to-end OSEDA flows have mostly been demonstrated on small designs and have not yet enabled large, industry-grade chips such as Linux-capable systems-on-chip (SoCs). This work presents Basilisk, the largest end-to-end open-source SoC to date. Basilisk’s 34 mm<sup>2</sup>, 2.7 MGE design features a 64-bit Linux-capable RISC-V core, a lightweight 124 MB/s DRAM controller, and extensive IO, including a USB 1.1 host, a video output, and a fully digital 62 Mb/s chip-to-chip (C2C) link. We implement Basilisk in IHP’s open 130 nm BiCMOS technology, significantly improving on the state-of-the-art (SoA) OSEDA flow. Our enhancements of the Yosys-based synthesis flow improve design timing and area by 2.3× and 1.6×, respectively, while consuming significantly less system resources. By tuning OpenROAD place and route (P&R) to our design and technology, we decrease the die size by 12%. The fabricated Basilisk chip reaches 62 MHz at its nominal 1.2 V core voltage and up to 102 MHz at 1.64 V. It achieves a peak energy efficiency of 18.9 DMFLOP/s/W at 0.88 V.

## I. INTRODUCTION

Many recently funded *Chips Acts* [1–3] aim at creating robust domestic silicon value chains, significantly increasing the number of chip designers, and reducing the cost and friction to access silicon manufacturing. One promising approach to realize these goals is *open-source silicon*: by sharing their designs under permissive open-source licenses, chip designers can freely collaborate and build on each other’s work. In fact, an *end-to-end* open design flow, shown in Fig. 2, would empower *anyone* to independently verify an entire chip step by step, enabling its safe use without requiring blind trust in the supply chain stakeholders. Thanks to a recent flourishing of open-source designs [4, 5], automation tools [6, 7], and process design kits (PDKs) [8], end-to-end open-source electronic design automation (OSEDA) flows [9, 10] already exist and produce viable results on smaller designs [11–14]. However, these open flows still lag behind their commercial closed-source counterparts and have not yet enabled designers to implement and optimize large, industry-grade designs such as application-class Linux-capable systems-on-chip (SoCs).

In this work, we present first silicon measurements of Basilisk<sup>1</sup>, to the best of our knowledge, the largest end-to-end open-source SoC to date. Basilisk is a Linux-capable 64-bit RISC-V SoC with a hierarchical interconnect, a lightweight 124 MB/s DRAM controller, and extensive IOs: UART, I2C, QSPI, a VGA video output, a four-port USB 1.1 host, and a fully digital 62 Mb/s chip-to-chip (C2C) link. We implemented Basilisk’s open 2.7 MGE design, fully based on open-source intellectual properties (IPs), in IHP’s open 130 nm BiCMOS PDK using OSEDA tools, significantly improving on the state-of-the-art (SoA) open flow in the process. We developed a capable open SystemVerilog (SV) frontend and optimized logic synthesis using Yosys, improving cell area by 1.6× and timing by 2.3× while reducing synthesis runtime by 2.5× and its memory usage by 4.8×. We further tuned place and route (P&R) in *OpenROAD* to our design, increasing core utilization by 10%. The fabricated 34 mm<sup>2</sup> chip reaches a clock frequency of 62 MHz, corresponding to 51 levels of logic (LL)<sup>2</sup>, at the nominal 1.2 V core voltage and up to 102 MHz at

1.64 V. On FP64 general matrix multiply (GEMM), we measure a peak energy efficiency of 18.9 MFLOP/s/W at the minimum operational core voltage of 0.88 V at 10 MHz.

## II. ARCHITECTURE

Fig. 1 shows Basilisk’s top-level architecture, which is based on our open-source, permissively licensed *Cheshire* [4] SoC platform and uses the RV64GC-compliant CVA6 [5] core. Basilisk integrates all hardware IPs needed to run Linux, including RISC-V-compliant interrupt controllers and a DRAM interface. To balance performance and area, Basilisk features a two-stage interconnect: high-throughput components attach to a fully connected 64-bit AXI4 [16] crossbar, while low-throughput peripherals attach to a lightweight Regbus [17] demultiplexer.

Basilisk’s fully digital HyperRAM controller supports two DRAM chips and transfer speeds of up to 124 MB/s. It connects to the AXI4 crossbar through a four-way, 473 MB/s, 64 KiB last-level cache (LLC), which can have each of its ways dynamically configured as a scratchpad memory. The CVA6 core is configured with four-way, 16 KiB L1 instruction and data caches.

Basilisk provides a rich set of peripherals. In addition to I2C, QSPI, and UART for serial communication, it includes a fully digital four-port 12 Mb/s USB 1.1 host controller and a VGA video output supporting XGA at 60 Hz. Each USB port is multiplexed with GPIOs, providing a software-controlled IO bus up to 8 bits wide. A JTAG test access point connected to a RISC-V debug module enables live debugging of the CVA6 core and full memory access. A fully digital duplex 62 Mb/s C2C link enables inter-chip communication through direct memory accesses. A high-efficiency 473 MB/s, asynchronous DMA engine [18] capable of 2D transfers can relieve CVA6 of data movement tasks. Basilisk’s peripherals are designed to be Linux-compatible, with many already having working drivers.

## III. OPEN-SOURCE IMPLEMENTATION

Industry-grade SV code presents a significant challenge for OSEDA toolchains. Existing approaches like SV2V [19] convert SV to simpler Verilog before elaboration. However, this bloats and obfuscates the hardware description, increasing synthesis runtime and memory usage while degrading the quality of results (QoR) [20]. To address this shortcoming, we developed *Yosys-Slang*<sup>3</sup>, an SV frontend for Yosys based on the leading open SV compiler *Slang* [21]. Unlike existing solutions, Yosys-Slang can directly elaborate industry-grade SV code and thus preserve design intent. This results in smaller netlists, lower memory usage, shorter runtime, better debuggability, and improved QoR.

We improved logic synthesis in Yosys. We added a specialized optimization mapping aligned shift operations to multiplexers and tuned the synthesis flow for our technology and design. We leveraged *lazy man’s synthesis* [22] to build a high-effort logic optimization script. Compared to a baseline synthesis flow derived from [9], our SV frontend and synthesis improvements reduce standard cell area from 1.8 to 1.1 MGE (1.6×) and increase post-synthesis clock frequency from 33 to 77 MHz (2.3×). They simultaneously reduce synthesis runtime from 5.4 to 1.7 h (3.2×) and its memory footprint from 35 to 7.4 GB (4.8×).

Finally, starting from the OpenROAD flow scripts (ORFS) [9],

<sup>x</sup> All authors contributed equally to this research.

<sup>1</sup>Basilisk source: <https://github.com/pulp-platform/cheshire-ihp130-0>

<sup>2</sup>Number of gates in longest path

<sup>3</sup>Yosys-slang source: <https://github.com/povik/yosys-slang>

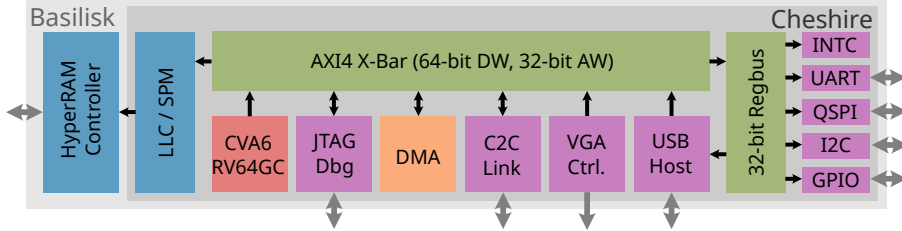


Fig. 1: Top-level architecture of Basilisk built on the Cheshire [4] SoC platform.

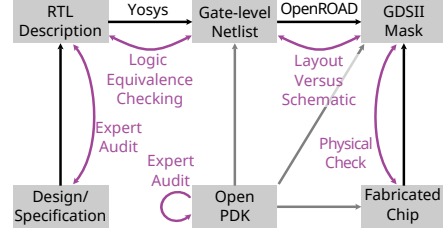


Fig. 2: End-to-end open-source chip design flow and verification of each step.

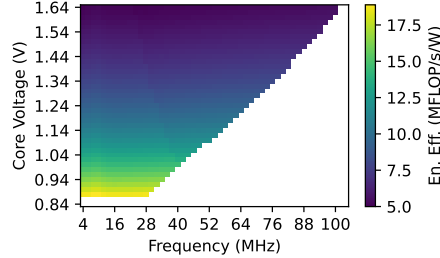


Fig. 3: Shmoo plot at 25 °C annotated with FP64 GEMM energy efficiencies.

	<i>Basilisk</i> [ours]	<i>MLEM</i> <sup>a</sup> [15]	<i>HEP</i> [11]	<i>Ghazi</i> [12]	<i>Ibtida</i> [13]	<i>GreenRio</i> [14]
Chip Area [ $\mu\text{m}^2$ ]	34.4	5.0	13.7	6.9	2.6	2.0
Chip Complexity [kGE]	2700	350	-	$\approx 500$	340	-
Std. Cell Complexity [kGE]	1140	102	230 <sup>b</sup>	$\approx 200$	94	53
Synthesis Speed [MHz]	77	80	25	12.5	-	80
Silicon Speed [MHz]	62	80	25	12.5	-	80
Logic levels (LL) <sup>c</sup>	51	58	-	-	-	-
Synth. Time [min]	130 <sup>d</sup> / 110 <sup>be</sup>	6.6 <sup>d</sup> / 65 <sup>be</sup>	-	-	-	6.2 / 120 <sup>e</sup>
Impl. Time [h]	24 <sup>d</sup> / 21 <sup>e</sup>	0.9 <sup>d</sup> / 8.8 <sup>e</sup>	-	-	-	2.0 / 38 <sup>e</sup>
Open PDK	IHP130	IHP130	$\times$	SKY130	SKY130	SKY130
Application-Class	$\checkmark$	$\times$	$\times$	$\times$	$\times$	$\times$

<sup>a</sup> Simple microcontroller reusing Basilisk’s flow <sup>b</sup> Assuming 50% placement density <sup>c</sup> Number of gates in longest path <sup>d</sup> 2.5 GHz Xeon E5-2670 <sup>e</sup> Normalized with standard cell complexity in MGE

TABLE I: Comparison to other open 130 nm designs implemented with open flows.

we tuned OpenROAD’s flow and hyperparameters in the global placement and routing stages to our design by better balancing routing and timing goals. We strategically added small routing blockages, especially around power grid connections, to guide local routing behavior. This significantly reduced routing congestion in dense modules like the boot ROM, allowing us to reduce the die area from 39 to 34 mm<sup>2</sup> (-12 %).

#### IV. SILICON RESULTS

Fig. 4 shows the fabricated Basilisk chip annotated with its final floorplan; the core area is dominated by the CVA6 core (45.6 %) and LLC (12.5 %). To characterize Basilisk, we created the evaluation board shown in Fig. 5, which breaks out its peripheral IO and provides the necessary circuitry for standalone operation. We use *DUTCTL* [23] to orchestrate on-chip workload execution with the external power supplies and clock sources.

We ran a 48×48 FP64 GEMM on Basilisk at 25 °C while sweeping the core voltage and frequency, resulting in the annotated Shmoo plot shown in Fig. 3. At the nominal 1.2 V voltage, Basilisk reaches a clock frequency of up to 62 MHz. At 1.64 V, we reach Basilisk’s peak frequency of 102 MHz. Basilisk reaches a competitive technology-normalized speed of 51 LL compared to another commercially implemented Cheshire-based design [4] at 46 LL. At the minimum operational core voltage of 0.88 V, we observe the peak energy efficiency of 18.9 MFLOP/s/W. Among ten packaged chips, we observe an average leakage power of 437  $\mu$ W.

Table I compares Basilisk to four other open 130 nm chips taped out using open design flows. Basilisk is the only application-class SoC among them, exceeding the next-largest design’s

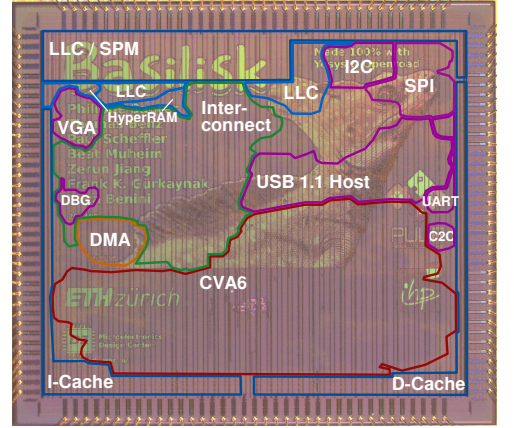


Fig. 4: Basilisk die shot and floorplan.

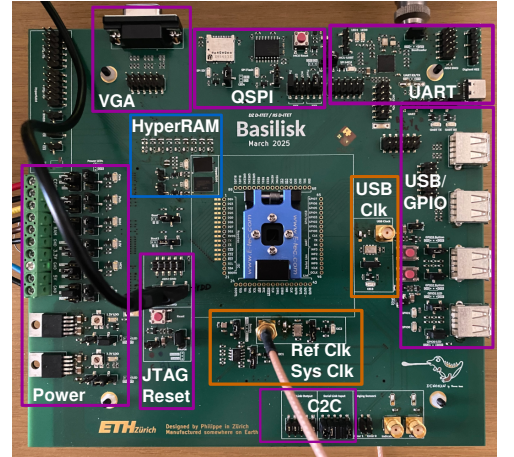


Fig. 5: Basilisk evaluation board.

standard cell complexity by 4.8× while maintaining competitive clock speeds. At 110 min/MGE and 21 h/MGE, Basilisk’s complexity-normalized synthesis and P&R runtimes outperforms *GreenRio*’s [14]. In fact, these runtimes are only bested by those of *MLEM* [15], a simple microcontroller design reusing Basilisk’s tuned OSEDA flow; this demonstrates the portability of our implementation flow improvements to other designs.

#### ACKNOWLEDGMENTS

We thank J. Schoenleber, A. Ottaviano, N. Wistoff, A. Prasad, C. Koenig, R. Balas, C. Reinwardt, N. Narr, and T. Senti. We also thank IHP for their generous support. Silicon manufacturing was sponsored by German BMBF project FMD-QNC (16ME0831).

#### REFERENCES

- [1] european-chips-act.com
- [2] congress.gov/bill/117th-congress/house-bill/4346
- [3] spectrum.ieee.org/indian-semiconductor-manufacturing
- [4] A. Ottaviano et al., *IEEE TCAS II*, 2023
- [5] F. Zaruba et al., *IEEE VLSI*, 2019
- [6] C. X. Wolf et al., *Austrochip*, 2013
- [7] T. Ajayi et al., *Proc. GOMACTECH*, 2019
- [8] K. Herman et al., *IEEE SSC Magazine*, 2024
- [9] github.com/The-OpenROAD-Project/OpenROAD-flow-scripts.
- [10] A. Ghazy et al., *WOSET*, 2020
- [11] T. Henkes et al., *DATE*, 2024
- [12] Z. R. Khan et al., *Authorea Preprints*, 2023
- [13] M. Khan et al., *techrxiv.16663738*, 2021
- [14] Y. Zhu et al., *WOSET*, 2022
- [15] P. Sauter et al., *arXiv:2502.05090*, 2025
- [16] A. Kurth et al., *IEEE Trans. Comput.*, 2021
- [17] github.com/pulp-platform/register\_interface. 2018
- [18] T. Benz et al., *IEEE Trans. Comput.*, 2023
- [19] github.com/zachj/sv2v. 2020
- [20] P. Sauter et al., *arXiv:2405.04257*, 2024
- [21] github.com/MikePopolski/slang. 2015
- [22] W. Yang et al., *IEEE/ACM ICCAD*, 2012
- [23] T. Benz et al., *RISC-V Summit Europe*, 2024