

Iguana: An End-to-End Open-Source Linux-capable RISC-V SoC in 130nm CMOS

Thomas Benz^{1*}, Paul Scheffler¹, Jannis Schönleber¹, Luca Benini^{1,2}

¹Integrated Systems Laboratory, ETH Zurich

²Department of Electrical, Electronic, and Information Engineering, University of Bologna

Abstract

Open-source architecture design and register-transfer level (RTL) descriptions, particularly around RISC-V, have made huge strides in the past decade. While open-source physical design and implementation are still lagging behind by comparison, they have recently been catching up: open EDA tools are nearing feature completeness, and some proprietary PDKs are being opened. We present Iguana, the first end-to-end open-source Linux-capable, 64-bit RV64GC, RISC-V System-on-Chip (SoC). Scheduled to tape out in IHP's 130nm technology, which is currently being open-sourced, Iguana sets important milestones for both open-source silicon and European silicon sovereignty. It implements our Cheshire architecture, which uses IP-based high-level synthesis (iHLS) to generate SoCs from carefully designed, silicon-proven open-source IPs. It includes a variety of standard peripherals, a DRAM controller, VGA display output, and a parallel die-to-die link. Cheshire, all its IPs and physical layers (PHYs) are released under a liberal Apache-based license. We implement Iguana with a fully open RTL-to-GDS toolchain, using established tools where possible and filling gaps with our open tools. Iguana is not a one-shot, but the first in a series of SoCs we will progressively extend and improve on. Furthermore, we have pre-validated our architecture through an FPGA mapping and a tested silicon prototype in TSMC's proprietary 65nm node.

Introduction

In recent years, open-source synthesis and physical implementation tools have significantly improved in quality [1], and some proprietary process design kits (PDKs) have been or are currently being open-sourced [2]. At the same time, the existing open-source architecture ecosystem around RISC-V has matured and is ready to be used in industry-grade silicon. Therefore, the time is ripe to work toward an *end-to-end open-source* Linux-capable RISC-V SoC.

Google and Efabless provide open tape-out runs in Skywater's 130nm node with their *Caravel* chip [3]. While this approach lowers the bar of entry for IC designers and provides a standard *smart* padframe, it comes with severe design restrictions that render larger SoCs with custom interfaces impossible. Submitted circuits cannot be larger than 10mm² or 1.5MGE and are to bound mandatory on-chip logic and a fixed padframe. This not only precludes custom IO in general, but also a DRAM interface typically necessary to run an operating system like Linux. Submitted designs are thus typically based around tiny 32-bit microcontrollers connected to custom accelerators.

Fully custom application-specific integrated circuits (ASICs) like Raven [4] have been manufactured, but lack the scale and features required to boot Linux autonomously. We present *Iguana*, a fully custom, end-to-end open-source Linux-capable, 64-bit RISC-V (RV64GC) SoC implemented in IHP's 130nm technology. In particular, we present the following contributions:

- We implement Iguana using a fully open-source tool pipeline from an industry-grade SystemVerilog register-transfer level (RTL) description to a GDS-II layout *without* source alterations to avoid tool limitations. Iguana will tape out in July using the European IHP's *SG13G2* technology [2].
- We build our SoC using the Cheshire [5] platform, which leverages IP-based high-level synthesis (iHLS) to generate arbitrary top-level architectures from hand-crafted silicon-proven open-source intellectual properties (IPs).
- We include two fully digital off-chip interfaces: HyperBus to provide our SoC with DRAM and a parallel die-to-die link to connect to other chips. Both interfaces, including their PHYs, are open-sourced using an Apache-based license.
- We provide various peripheral IO on Iguana, including UART, SPI, I2C, GPIOs, a VGA display output, and JTAG for live debugging.
- We verify Iguana and our Linux boot flow through an FPGA implementation and a tested silicon prototype in TSMC's proprietary 65nm node.

Iguana is the first ASIC in a series of three IHP 130nm tape-outs planned for 2023.

Architecture

Iguana is built around the *Cheshire* platform, a modular framework to create custom Linux-capable SoCs based around CVA6 [6], a 64-bit RV64GC core. Cheshire uses an iHLS approach, assembling systems from a pool of carefully designed silicon-proven IPs

*Corresponding author: tbenz@iis.ee.ethz.ch

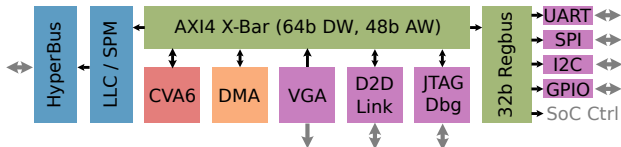


Figure 1: Block diagram of the Iguana SoC.

instead of generating all RTL and impeding interpretability and verification.

Iguana, shown in Figure 1, is built around a central AXI4 crossbar allowing the CVA6 core to communicate with the rest of the system. Efficient bulk memory movement is implemented using a transfer-descriptor-based DMA engine. The off-chip HyperBus memory controller is connected to our eight-way AXI4 last-level cache (LLC), of which each way can be configured either as L2 scratchpad memory (SPM) or cache.

Our parallel die-to-die link can be used for bidirectional communication either with another Iguana ASIC or an FPGA. Iguana provides various peripheral IPs, including UART, I2C and SPI hosts, and GPIOs. A VGA controller allows for display output. A RISC-V-compliant JTAG debug module is present, aiding the debug process. Iguana supports boot through preloading over JTAG or UART, and stand-alone boot from SD card, SPI NOR flash, or I2C EEPROM.

RTL-to-GDS Toolchain

Iguana is synthesized and implemented using a fully open-source RTL-to-GDS-II flow. We employ established tools, Yosys for synthesis and OpenROAD for implementation, wherever possible. The tools are orchestrated through TCL scripts, an established approach in industry-grade EDA toolflows. We fill any gaps in our toolchain that cannot be filled with existing solutions with our open-source tools. We use *Bender* [7], an IP management and dependency resolution tool, to collect our IPs. *Morty* and *SVase* resolve macros and simplify language constructs not supported by downstream tools. The RTL is translated to pure verilog using *sv2v* and read into Yosys for synthesis. The backend implementation is done using tools from OpenROAD. All our contributions to the open-source tools will be upstreamed to benefit the entire community. Unlike current approaches, our flow does not require the RTL to be manually rewritten/converted to a limited subset of SystemVerilog, removing the barrier of entry for industry-grade IPs.

Once the PDK is fully open, we plan on seizing the unique opportunity to release all used scripts, sharing years of tape-out know-how from our organization that we had to hide so far with the open source community.

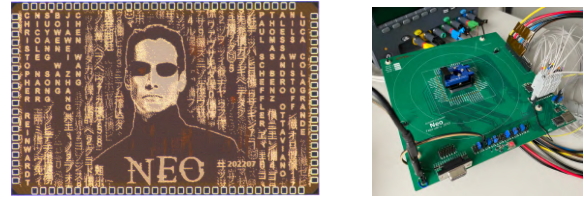


Figure 2: (left) die shot of Neo, (right) Neo mounted using a quick-release socket on a test PCB.

Silicon Demonstrator

Iguana’s architecture is silicon-proven and verified through a previous tape-out, Neo, which was implemented using a closed-source flow and industry-grade tools in TSMC’s 65nm node. Neo achieves a clock frequency in excess of 325 MHz at 1.2 V. Figure 2 shows a die shot and bring-up board of Neo; its stand-alone operation and peripherals are tested and working.

Conclusion and Outlook

We present Iguana, the first end-to-end fully open-source, Linux-capable, 64-bit RV64GC RISC-V SoC. Iguana is, after Neo, the second chip using the new Cheshire platform to implement a Linux-capable SoC. We use a fully custom flow proven in hundreds of successfully taped-out ASICs and established open-source synthesis and implementation tools. Targeting a European open-source PDK, Iguana is a significant step toward true European silicon sovereignty. It is planned to tape out early July 2023 in the IHP 130nm node. It is then followed by *Tegu* in September and *Komodo* in December, both deploying major architectural and design innovations: aging sensors, real-time interconnect, multicore coherent CVA6, or side-channel prevention.

References

- [1] T Ajayi et al. “OpenROAD: Toward a Self-Driving, Open-Source Digital Layout Implementation Tool Chain”. In: *Proc. GOMACTECH* (2019), pp. 1105–1110.
- [2] IHP-GmbH. *IHP Open Source PDK*. <https://github.com/IHP-GmbH/IHP-Open-PDK>. 2022.
- [3] efabless. *Caravel Harness*. <https://github.com/efabless/caravel>. 2022.
- [4] efabless. *RAVEN*. https://platform.efabless.com/design_catalog/asic_platform/116. 2018.
- [5] Alessandro Ottaviano et al. *Cheshire: A Lightweight, Linux-Capable RISC-V Host Platform for Domain-Specific Accelerator Plug-In*. 2023. arXiv: 2305.04760 [cs.AR].
- [6] Florian Zaruba and Luca Benini. “The Cost of Application-Class Processing: Energy and Performance Analysis of a Linux-Ready 1.7-GHz 64-Bit RISC-V Core in 22-nm FD-SOI Technology”. In: *IEEE Transactions on VLSI Systems* (2019).
- [7] PULP Platform. *bender*. <https://github.com/pulp-platform/bender>. 2022.