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

Integrated Systems Laboratory (ETH Zürich)

Thomas Benz  
tbenz@iis.ee.ethz.ch  
Paul Scheffler  
paulsc@iis.ee.ethz.ch  
Jannis Schönleber  
janniss@iis.ee.ethz.ch

PULP Platform
Open Source Hardware, the way it should be!
Situation and Challenges

• **FOS RTL established**
  • Led to a major increase in hardware research output

• **FOS Synthesis + Backend**
  • Under development
  • Will be the **next frontier**

• **Fully open hardware?**
  • Caravel (Skywater) 10 mm²
  • Small designs, limited IO

• **What about European solutions?**
Our Contributions

• Implementing Iguana in IHP’s 130nm node
  • Tapeout through Europractice in July 2023
  • First end-to-end FOS ASIC capable of running Linux
  • Custom padframe allowing us full control over IO
  • IHP’s open European PDK and fab

• Building Iguana from industry-grade IPs
  • Cheshire SoC framework
  • Including 2 fully digital off-chip interfaces
    • HyperBus off-chip DRAM interface
    • Chip-to-chip link

• Providing full peripheral IO → Desktop Linux minicomputer
• Verifying Linux boot through FPGA and silicon demonstrator
Architecture

- **Goal**: Linux Desktop computer
- **CVA6**: RV64GC
- **AXI4 and Regbus used in interconnect**
- **AXI4-based last-level cache / SPM**
- **Peripherals**
- **Standalone boot**
  - SPI (SD Card – GPT)
  - I2C
- **FOS, digital-only, PHYs**
  - HyperBus off-chip memory interface
  - Chip-2-Chip link
The Cheshire Concept & Silicon Demonstrator

- **Iguana is built using Cheshire**
  - Silicon-proven Linux SoC framework, FPGA port
  - Parametrizable top
  - iHLS (IP-based high-level synthesis)
    - Template-based assembly of parametrizable IPs
    - Complex top-levels

- **Silicon Demonstrator**
  - Neo tapeout in 2021
  - TSMC 65nm node, closed toolchain
  - Similar SoC, different DRAM controller
  - Tested and is working standalone 😊
Our Flow: In-house Tools

- **Bender**
  - Source management
  - Script generation
  - Similar to FuseSoC
  - Resolve project dependencies

- **Morty**
  - Source pickler → single file, single context
  - Macro expansion

- **SVase**
  - Parameter and generate pre-elaboration
  - Human-readable simplification of SV
  - Uses the *slang* SystemVerilog parser

Additional Resources:
- [Bender](https://github.com/pulp-platform/bender)
- [Morty](https://github.com/pulp-platform/morty)
- [SVase](https://github.com/pulp-platform/svase)
Frontend: RTL to Netlist

- Bender and Morty to handle sources
- SVase: parameter elaboration & simplification
- SV2V to transform the remaining SV constructs
- Yosys synthesis

github.com/MikePopoloski/slang
github.com/zachjs/sv2v
github.com/YosysHQ/yosys
Backend

- OpenRoad and Klayout

- Custom TCL-only flow
  - Based on our traditional flow called “cockpit”
  - Simpler, in-line with our teaching
  - Inspired by the OpenRoad flow scripts

- Top-down design hierarchy
  - Most of the area is occupied by CVA6

- High turnaround time of ~33h
  - Many steps are single-threaded
Results

- Contribution to the FOS Flow
  - Bugfixes and improvements to the tools
  - Demonstrate a complex FOS design
  - Improve IHP’s open standard cells
    - Hackathon at ETH
    - Cooperation with HS RheinMain

- Targeting fully-open GDS
  - SRAM macros and I/O cells are still closed
    - Opening imminent
  - 40mm² in IHP 130nm
  - > 50 MHz (WC, conservative)
Conclusion and Outlook

• Linux-capable RV64GC RISC-V SoC
• FOS off-chip DRAM and chip-2-chip link
• Industry-grade SystemVerilog IPs
• OpenRoad backend with “cockpit” flow
• Tapeout through Europractice in July 2023
  • First end-to-end FOS Linux-capable ASIC
  • Establish FOS flow for complex designs
• Future Tapeouts are planned
  • Tegu and Komodo: will carry scientific work loads
  • Multicore CVA6, real time SoC, side channel prevention

2023 RISC-V Summit Barcelona