Pacific Northwest NATIONAL LABORATORY

POLITECNICO MILANO 1863

HARVARD

ASU

From Python to Silicon: First Tapeouts Produced by an End-to-End Open-Source Hardware Compiler

June 27, 2026

Ankur Limaye, Nicolas Bohm Agostini, David Kong, Nrusinga Charan Gantayat, Gianmarco Accordi, Max Ramstad, Lakshmi Varshika Mirtinti, Vito Giovanni Castellana, Joseph Manzano, Jeff (Jun) Zhang, Gage Hills, Fabrizio Ferrandi, **Antonino Tumeo**

U.S. DEPARTMENT OF ENERGY BATTELLE

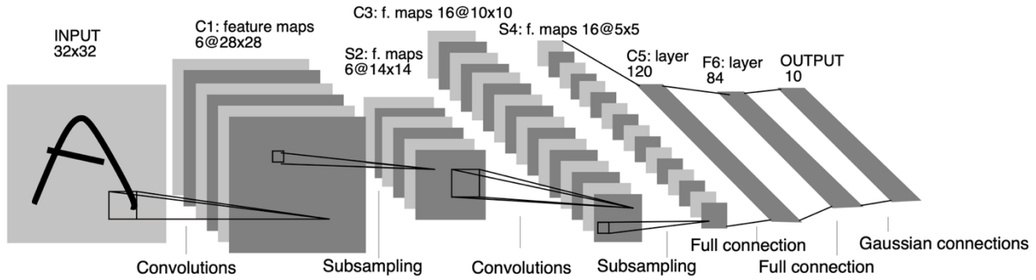
PNNL is operated by Battelle for the U.S. Department of Energy

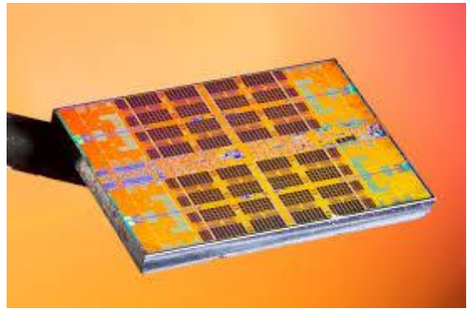
Motivations

- Data science algorithms, approaches, and frameworks are quickly evolving
- Domain-specific accelerators are the only possible approach to keep increasing performance in tight constraints
- Existing accelerators start from specific models (i.e., mostly deep neural networks) or only try to accelerate specific computational patterns coming from high-level frameworks
- Designing hardware by hand is complex and time-consuming
- Depending on the application, a designer may want to explore performance, area, energy, accuracy, and more…

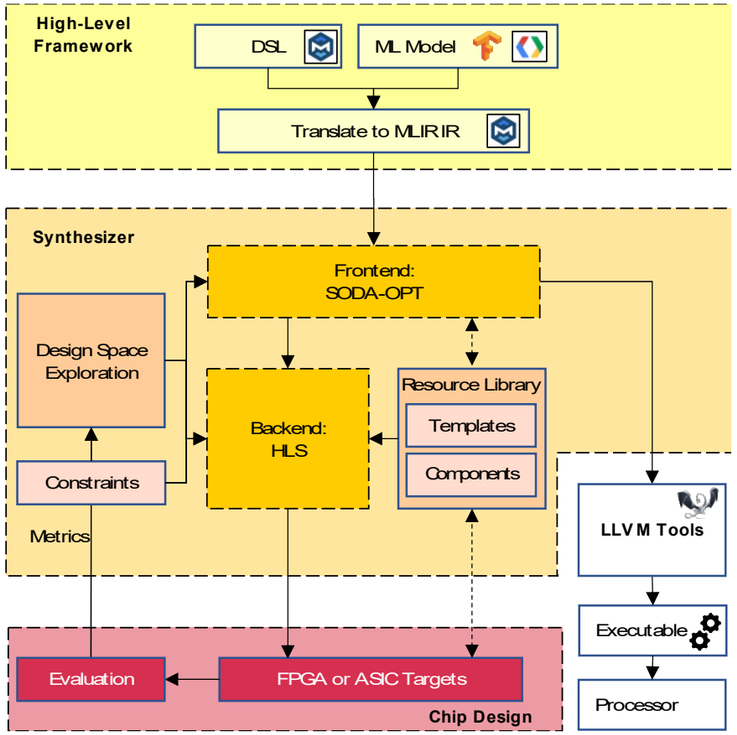
- ***Need tools to quickly transition from formulation of an algorithm to the accelerator implementation and explore the accelerator design along different dimensions***

LeNet architecture from the original paper





SODA Synthesizer: Overview



- A modular, multi-level, interoperable, extensible, **open-source hardware compiler** from **high-level programming frameworks to silicon**
- Compiler-based frontend, leveraging the MultiLevel Intermediate Representation (MLIR)
- **Compiler-based backend**, leveraging state-of-the-art High-Level Synthesis (HLS) techniques
- Generates **synthesizable Verilog** for a variety of targets, from Field Programmable Gate Arrays (FPGAs) to Application Specific Integrated Circuits (ASICs)
- Optimizations at all levels are performed as **compiler optimization** passes

[M. Minutoli, V. G. Castellana, C. Tan, J. Manzano, V. Amatya, A. Tumeo, D. Brooks, G-Y. Wei: SODA: a New Synthesis Infrastructure for Agile Hardware Design of Machine Learning Accelerators. ICCAD 2020: 98:1-98:7]

[J. Zhang, N. Bohm Agostini, S. Song, C. Tan, A. Limaye, V. Amatya, J. Manzano, M. Minutoli, V. G. Castellana, A. Tumeo, G-Y. Wei, D. Brooks: Towards Automatic and Agile AI/ML Accelerator Design with End-to-End Synthesis. ASAP 2021: 218-225]

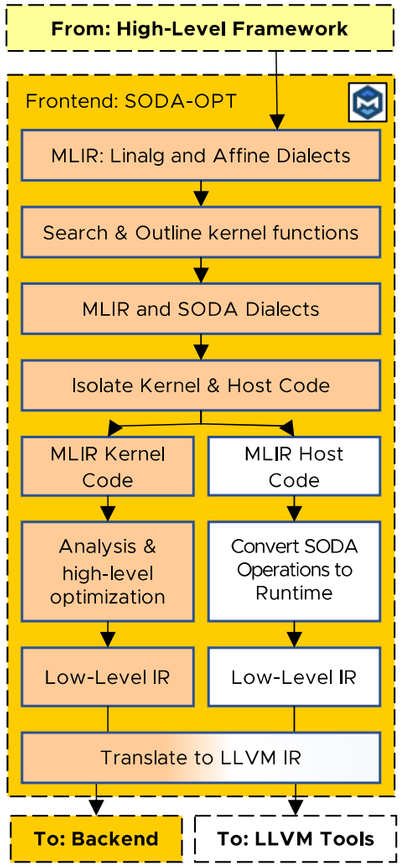
[N. Bohm Agostini, S. Curzel, J. Zhang, A. Limaye, C. Tan, V. Amatya, M. Minutoli, V.G. Castellana, J. Manzano, D. Brooks, G-Y. Wei, A. Tumeo: Bridging Python to Silicon: The SODA Toolchain. IEEE Micro Magazine 2022 - **Best Paper for 2022]**

[N. Bohm Agostini, S. Curzel, V. Amatya, C. Tan, M. Minutoli, V. G. Castellana, J. Manzano, D. Kaeli, A. Tumeo : An MLIR-based Compiler Flow for System-Level Design and Hardware Acceleration. ICCAD 22]

[Fabrizio Ferrandi, Vito Giovanni Castellana, Serena Curzel, Pietro Fezzardi, Michele Fiorito, Marco Lattuada, Marco Minutoli, Christian Pilato, Antonino Tumeo: Invited: Bambu: an Open-Source Research Framework for the High-Level Synthesis of Complex Applications. DAC 2021: 1327-1330]

SODA-OPT: Frontend and High-Level IR

- **SODA-OPT: S**earch, **O**utline, **D**ispatch, **A**ccelerate frontend **opt**imizer “generates” the SODA High-Level IR
- Employs and embraces the MLIR framework
  - MLIR: Multi-Level Intermediate Representation
  - Used in TensorFlow, TFRT, ONNX-MLIR, NPComp, others
  - Several architecture independent dialects (Linalg, Affine, SCF) and optimizations
- Interfaces with high-level ML frameworks through MLIR “bridges” (e.g., libraries, rewriters)
- Defines the SODA MLIR dialect and related compiler passes to:
  - Identify dataflow segments for hardware generation
  - Perform high-level optimizations (dataflow transformations, data-level and instruction-level parallelism extraction)
  - Generate interfacing code and runtime calls for microcontroller



SODA-OPT: System Overview

https://github.com/pnnl/soda-opt

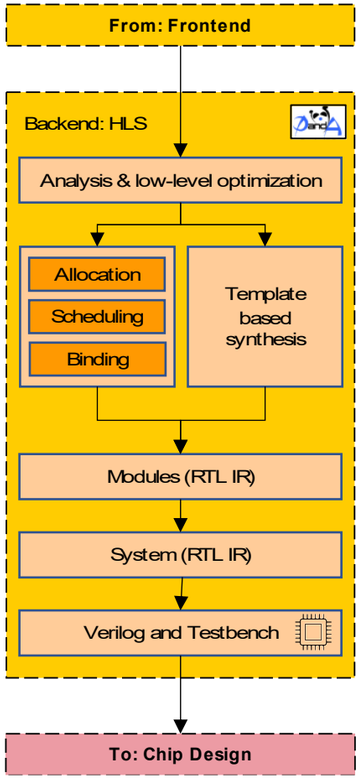
[N. Bohm Agostini, S. Curzel, V. Amatya, C. Tan, M. Minutoli, V. G. Castellana, J. Manzano, D. Kaeli, A. Tumeo : An MLIR-based Compiler Flow for System-Level Design and Hardware Acceleration. ICCAD 22]

[N. Bohm Agostini, S. Curzel, D. Kaeli, A. Tumeo: SODA-OPT an MLIR based flow for co-design and high-level synthesis. CF 2022: 201-202 - **Best Poster Award**.]

Pacific Northwest NATIONAL LABORATORY

SODA Synthesizer: HLS Backend

- The synthesizer backend take as input the properly optimized low-level IR and generate the hardware descriptions of the accelerators
- The HLS backend is PandA-Bambu, an open-source state-state-of-the-art high-level synthesis (HLS)
  - Key features: **parallel accelerator designs**, **modular HLS**, and **ASIC support**
- The HLS backend provides automated testing and verification of the generated designs



Bambu HLS

https://panda.dei.polimi.it

[Fabrizio Ferrandi, Vito Giovanni Castellana, Serena Curzel, Pietro Fezzardi, Michele Fiorito, Marco Lattuada, Marco Minutoli, Christian Pilato, Antonino Tumeo: Invited: Bambu: an Open-Source Research Framework for the High-Level Synthesis of Complex Applications. DAC 2021: 1327-1330]

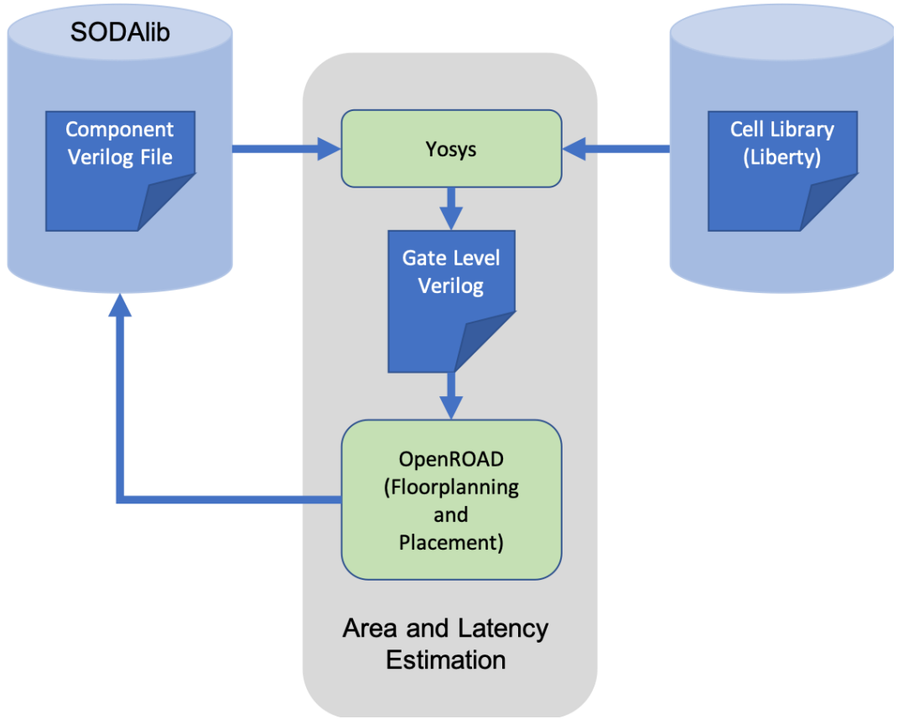
SODA Synthesizer: ASIC targets

- The multi-level approach of the SODA toolchain allows supporting different target technologies (FPGA, ASIC) for actual generation of the designs

- SODA also supports ASIC targets:
  - **Commercial Tools** (Synopsys Design Compiler with Global Foundries 12/14 nm cells)
  - **OpenROAD suite** (OpenPDK 45nm and ASAP 7nm cell libraries)

- Backends' resources characterized for the target technology:
  - **HLS Backend: Eucalyptus** tool in Bambu, allows driving hardware synthesis algorithms to optimize for area, latency, etc.

- PandA-Bambu now also the opensource C frontend for **ZeroASIC' SIliconCompiler** (https://www.siliconcompiler.com)

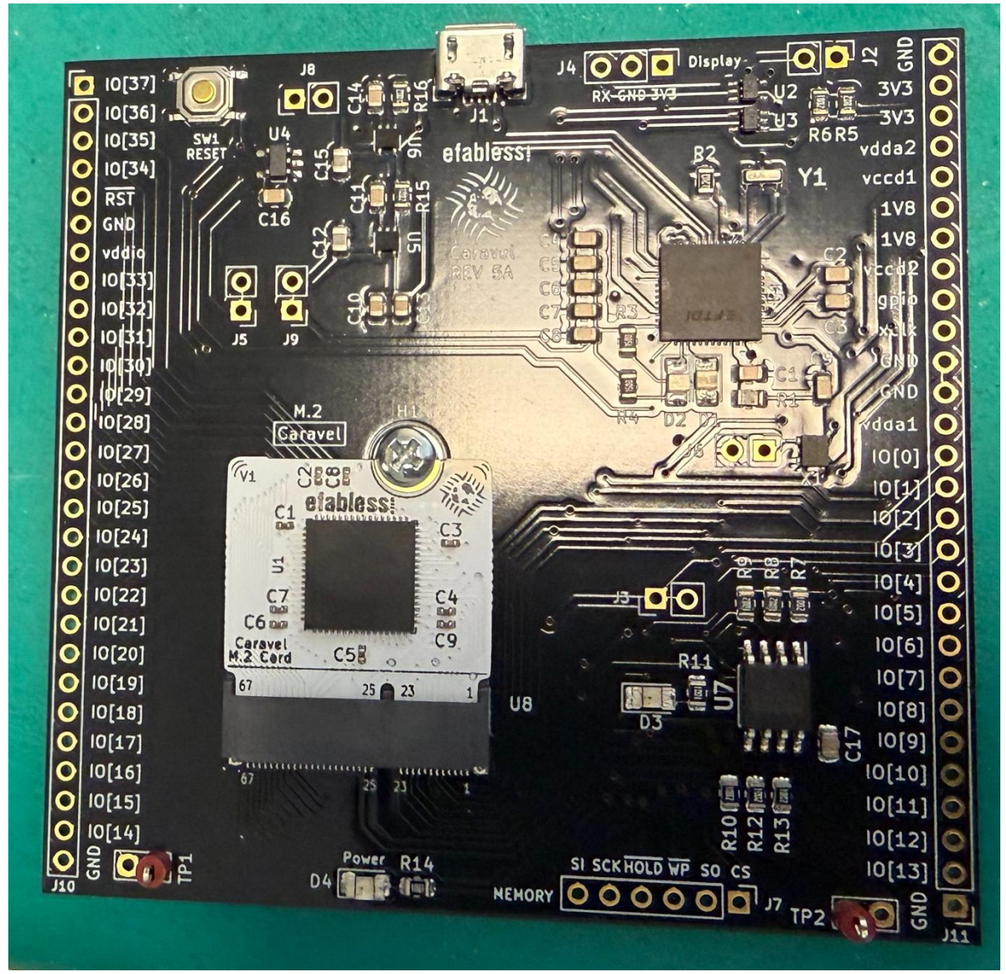


SODA characterization flow. The characterization flow can be extended to synthesize HLS generated designs, or used to estimate their area-latency-power profiles to drive the Design Space Exploration engine

OpenROAD
https://theopenroadproject.org

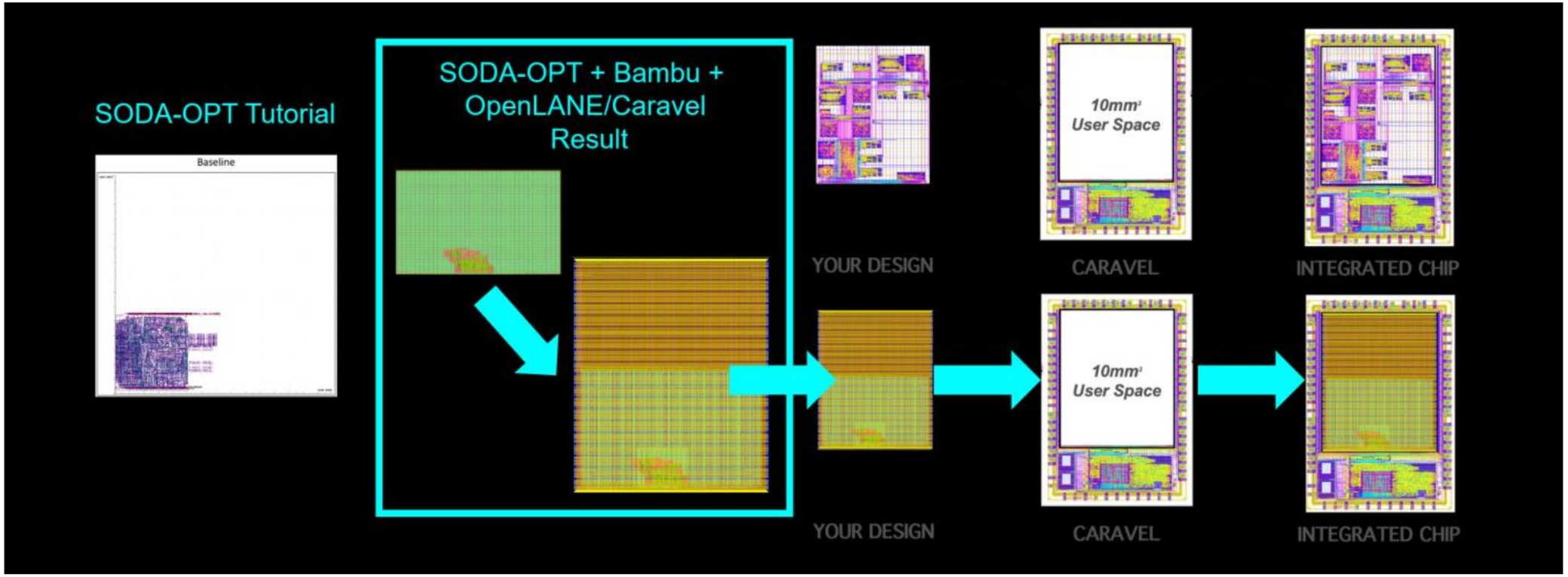
Literally, from Python to Silicon, only with Open-source Tools (with Harvard University)

- eFabless flow (before becoming Chipfoundry)
- Uses SkyWater 130 nm open process development kit (PDK)
- Uses OpenLane (repackaged OpenROAD flow) for physical Design and layout
- Implements exactly the baseline, unoptimized accelerator of the SODA tutorial
  - Only modification: insert SRAM macro between harness platform microcontroller (PicoRV) and wishbone
  - Bambu can generate also wishbone interfaces
- Exploration of design, interface integration, and submission of tapeout executed by Harvard University



Pacific Northwest
NATIONAL LABORATORY

Conceptual Flow



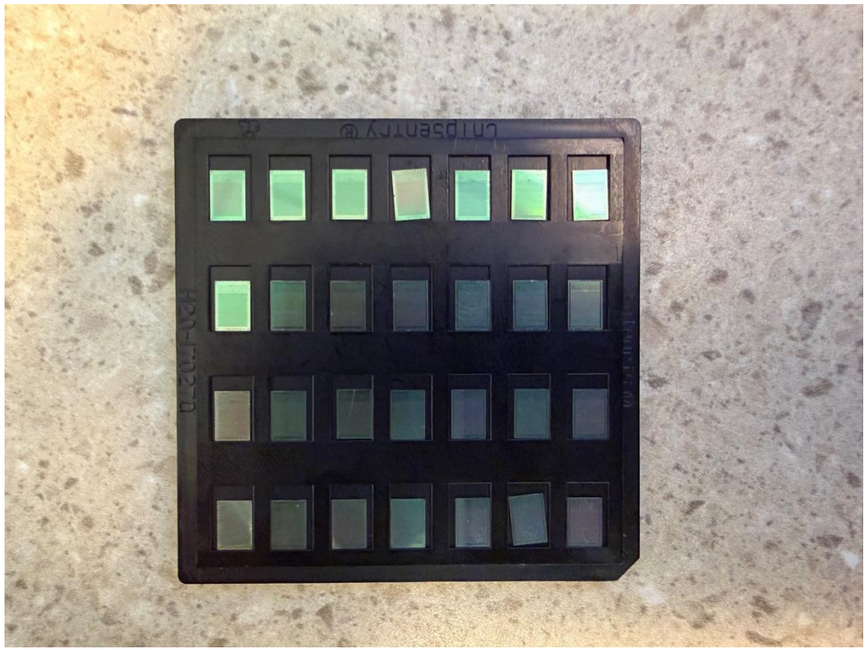
Pacific Northwest
NATIONAL LABORATORY

Technical Details

- Caravel (harness including IO pad with a PicoRV processor) runs at 40 MHz and provides a Wishbone bus interfaces
- Desing integrates a single ported SRAM IP of 4 KB, wrapped into wishbone
  - SRAM can run at 100 MHz, so it can be turned as a dual port by running it in a DLL (Delay-Locked Loop)
- Currently, IO pad and PicoRV confirmed working (simple test through firmware that makes led blink through GPIO port)
- Next steps:
  - Access memory (slave mode) from processor, and read back data
  - Write to memory triggering accelerator start, read back data
- Limitation on single serial port controller acting both as JTAG and device output

Two Other Tapeouts: New ChipFoundry Tapeout (delivered)

- First ChipFoundry shuttle
- Again SkyWater 130 nm
- Similar design from SODA tutorial (unoptimized batched_matmul operator)
- However, does not include SRAM
  - Testing process expected to be more complicated, with need to leverage combination of processor and pattern generators
- However, frequency increased to 50 MHz
- Tapeout delivered, contains: 10 chips packaged and soldered on M2 daughter board, 100 chips packaged in QFN64, and 53 bare dies



Two Other Tapeouts: GF180 (with ASU)

- Global Foundries 180 nm + Synopsys academic pilot
  - In collaboration with Arizona State University
- Accelerator design is much more complicated: encoder part of an autoencoder used for denoising with the energy electron loss spectroscopy (EELS) material characterization method
- However, no SRAM IPs available, no memory compiler, and IO pad does not host a processor
- IO pad has only 18 usable GPIO pins
- Accelerator is wrapped in a module to perform CRC check and verify that operations are being performed (load/store, transitions of the state machine)
- Need to decide on packaging
- Need to design/reuse PCB for testing

Plans for Next Tapeout Demonstration

- Targeting ChipFoundry September Shuttle (CI2609)
- Accelerator still targeting machine learning
  - Considering one of the TinyML examples
- Improving **physical awareness** of the high-level synthesis process by:
  - Characterizing resources (functional units) targeting the actual open-source SkyWater 130 nm PDK
  - Implementing scheduling algorithms in the HLS tool to take into account wire delay
  - Improving how the interconnection step is performed (currently optimized for FPGAs rather than ASIC)
- Considering opportunities for Design for Testability
  - E.g., adding CRC check and/or scan chains in front of the accelerator
- Dates:
  - Commitment: July 18 2026
  - Tapeout: September 16 2026
  - Delivery: March 3 2027

Pacific Northwest NATIONAL LABORATORY

Lessons learned so far

- Open-source tools are mature enough for supporting a tapeout. However, HLS approaches still very FPGA oriented
- Proprietary tools (Synopsys applied for GF180) still give better quality of results and timing closure opportunities
- Especially in the context of prototyping, testing should not be an after thought
- Open-source PDKs enable sharing results and feedbacks…
  - This is critically important for AI-driven design and verification methods
- …but are still missing important components
  - Memory macros and/or support for a memory compiler are missing
- GF180 and Sky130 nodes are more relevant for analog design than for digital
  - Need for more advanced nodes for digital design
- Packaging decisions are also extremely relevant, and if moving to a chiplet-based world, there is an opportunity for developing new packaging-aware design methodologies

Public Software Repositories

- SODA-OPT: https://github.com/pnnl/sodaopt
- SODA-Benchmarks: https://github.com/pnnl/soda-benchmarks
- Panda-Bambu HLS: https://panda.dei.polimi.it (latest release 2024.10)
  - Next release will move to APACHE 2.0 license (already available in release_candidate branch)
- OpenROAD: https://theopenroadproject.org (external tool, leveraged by SODA toolchain to achieve end-to-end synthesis to ASIC in a fully opensource compiler toolchain)
- SODA docker image: https://hub.docker.com/r/agostini01/soda

SODA-OPT

SODA-Benchmarks

PandA-Bambu HLS (2024.10)

SODA Docker Image

SODA Tutorial: *ISCA 2026 (this morning)*

Pacific Northwest
NATIONAL LABORATORY

Conclusions

- SODA implements an **end-to-end** (high-level frameworks to silicon) **compiler-based toolchain** for the generation of domain-specific accelerators
  - Modular, multi-level, extensible
  - All based on interoperating open-source technologies
  - Targets reconfigurable architectures FPGAs as well ASICs
  - Considers system-level implications
  - Enables automated design space exploration and agile hardware design
- We now have silicon demonstration of the full design pipeline
  - Linear algebra benchmark from the tutorial
  - DOE-relevant accelerator
  - SODA is independent of PDKs and use of commercial/open-source backend tool
- The open-source chip flows are fully reproducible and the post-silicon characterization witll provide silicon-validation of SODA's performance, power, and timing-model accuracy, directly benefiting the open-source hardware design community

IEEE Computer Magazine Special Issue

- **Democratizing Microelectronics: Open-Source Tools and Chiplet-Enabled Prototyping**
  - Special Issue wants to capture the transformation enabled by Open-Source Hardware Design Tools (from architectural simulators, to generators, to prototyping platforms and electronic Design Automation) and Modular Chiplet-based Design Methodologies
  - Targets prototyping as a way to prove novel hardware designs concepts
- Guest Editors: Antonino Tumeo (PNNL), Andrew Kahng (UCSD), Cristina Silvano (Politecnico di Milano), Jeffrey Voas (NIST)
- Deadline: August 1, 2026
- https://www.computer.org/digital-library/magazines/co/cfp-democratizing-microelectronics-open-source-chiplet

Pacific Northwest
NATIONAL LABORATORY

Thank you!

- This work has been partially supported by:
  - The DOE-SC project DeCoDe (Democratizing Co-design), part of the MEERCAT Microelectronics Science Research Center (MSRC)
  - The DOE ASCR Competitive Portfolio Project ENCODE (End-to-end Co-design)
  - The DOD Two-Truths Project

- Questions?
  - antonino.tumeo@pnnl.gov
  - vitogiovanni.castellana@pnnl.gov