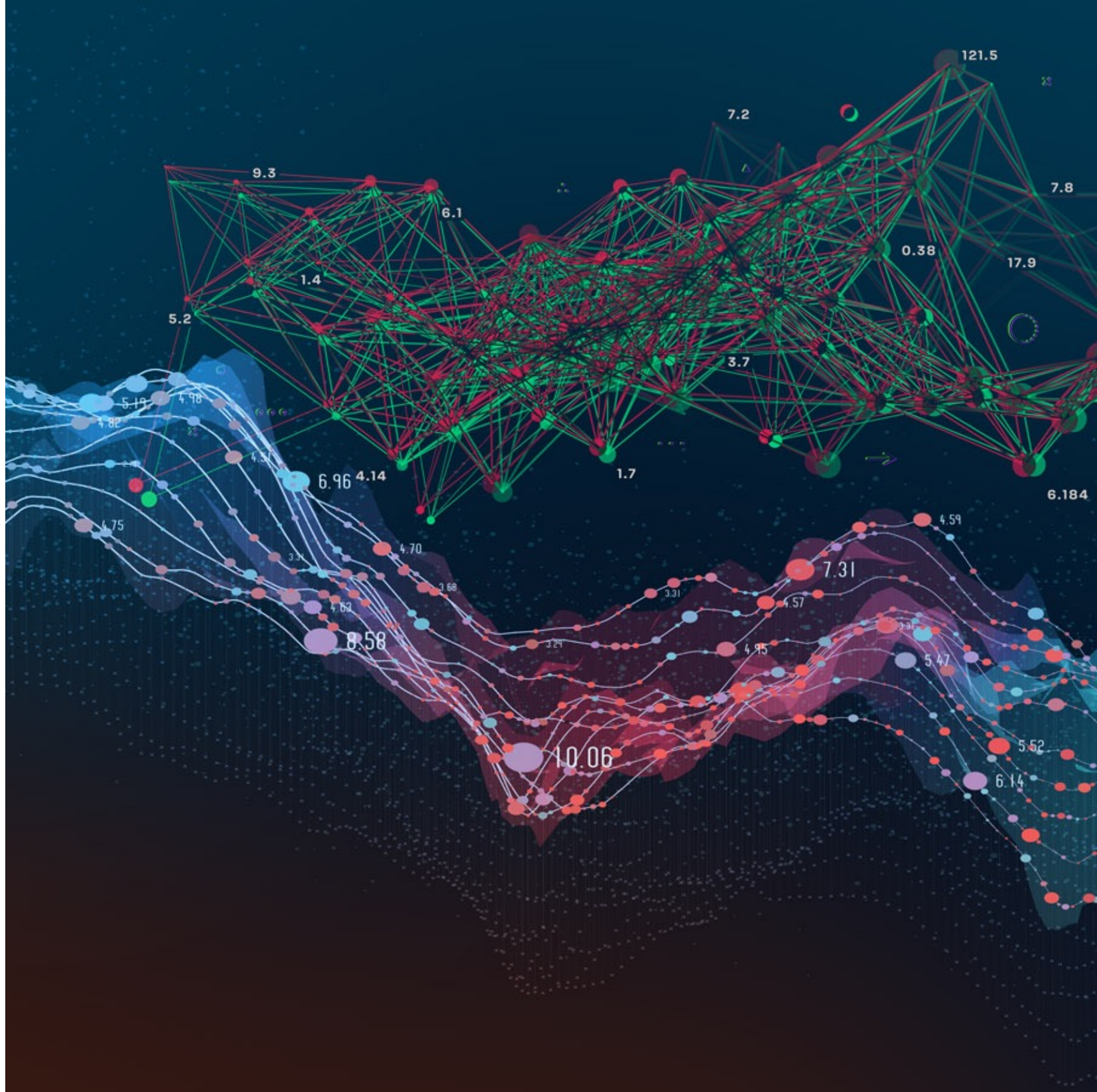
Pacific Northwest
NATIONAL LABORATORY

SODA Synthesizer: an Open-Source, End-to-End Hardware Compiler

Nicolas Bohm Agostini, Serena Curzel,
Ankur Limaye, Marco Minutoli, Vito Castellana,
Joseph Manzano, Fabrizio Ferrandi, Antonino Tumeo

Antonino Tumeo

Chief Scientist, High Performance Computing Group



U.S. DEPARTMENT OF ENERGY BATTELLE

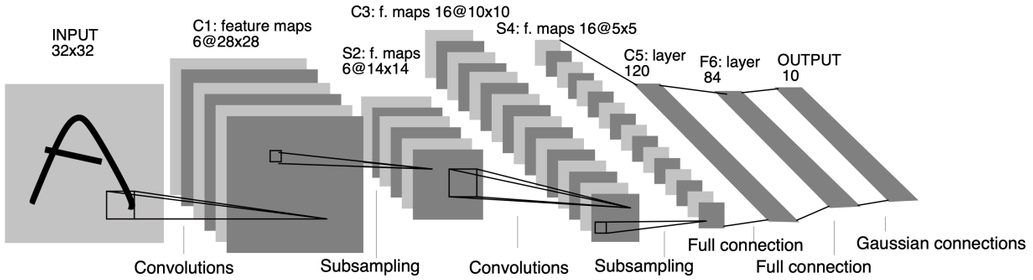
PNNL is operated by Battelle for the U.S. Department of Energy

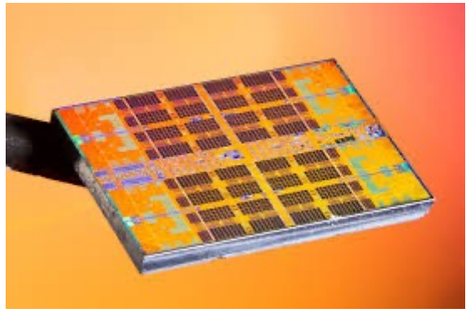
Motivations

- 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…

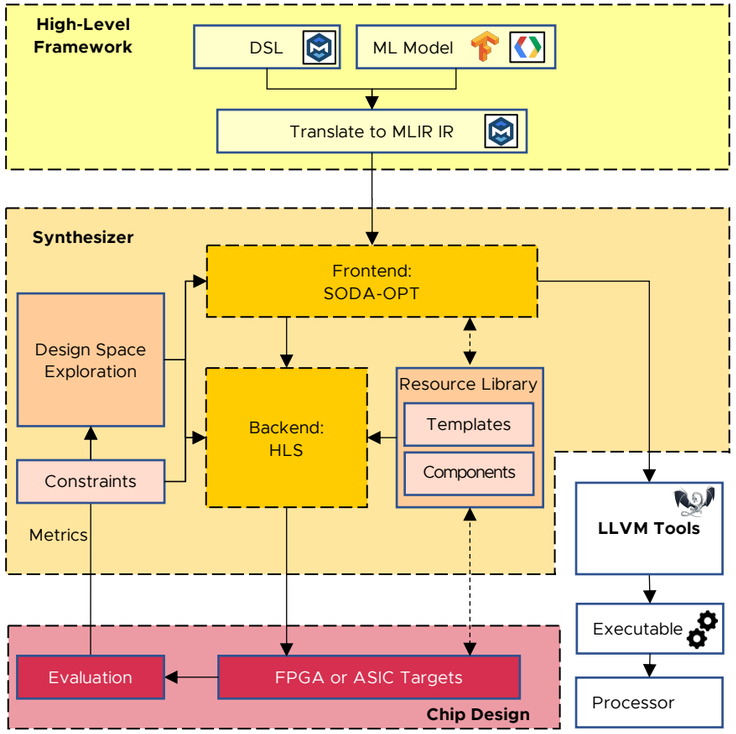
- ***Need tools to quickly transition from formulation of an algorithm to the accelerator implementation and explore the accelerator design along different dimensions***

LeNet architecture from the original paper





SODA Synthesizer: Overview



[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]

- 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, as well as a Coarse-Grained Reconfigurable Array (CGRA) generator
- 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

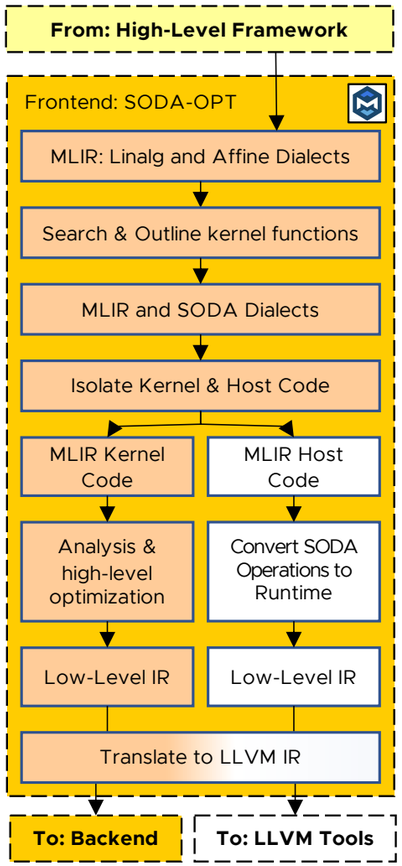
[N. Bohm Agostini, S. Curzel, J. Zhang, A. Limaye, C. Tan, V. Amatya, M. Minutoli, V.G. Castellana, J. Manzano, A. Tumeo: Bridging Python to Silicon: The SODA Toolchain. IEEE Micro Magazine, to appear.]

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

[N. Bohm Agostini, D. Kaeli, A. Tumeo: SODA-OPT: System-Level Design in MLIR for HLS. SC 21 Poster]

[N. Bohm Agostini, S. Curzel, V.C. 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 2022]



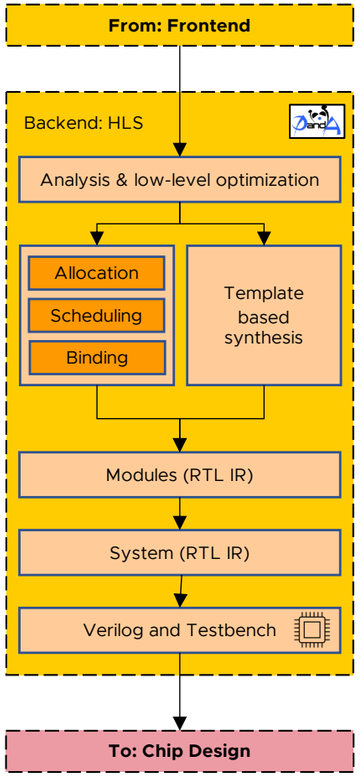
SODA-OPT: System Overview

https://gitlab.pnnl.gov/sodalite/soda-opt

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
- Note: SODA-OPT now also supports output to commercial HLS tools (Vits-HLS)



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]

Pacific Northwest
NATIONAL LABORATORY

Why an HLS Backend?

- Provides the necessary **generality** to deal with novel algorithms
- Provides opportunities for **specialized** and optimized templates by recognizing specific computational patterns

- The SODA Approach relies on **progressive lowerings** of compiler intermediate representations (IRs), rather than rewriting annotated C/C++
  - Reduces semantic mismatches between high-level and low-level descriptions
  - Provides further opportunities to apply optimizations at the right level of abstraction

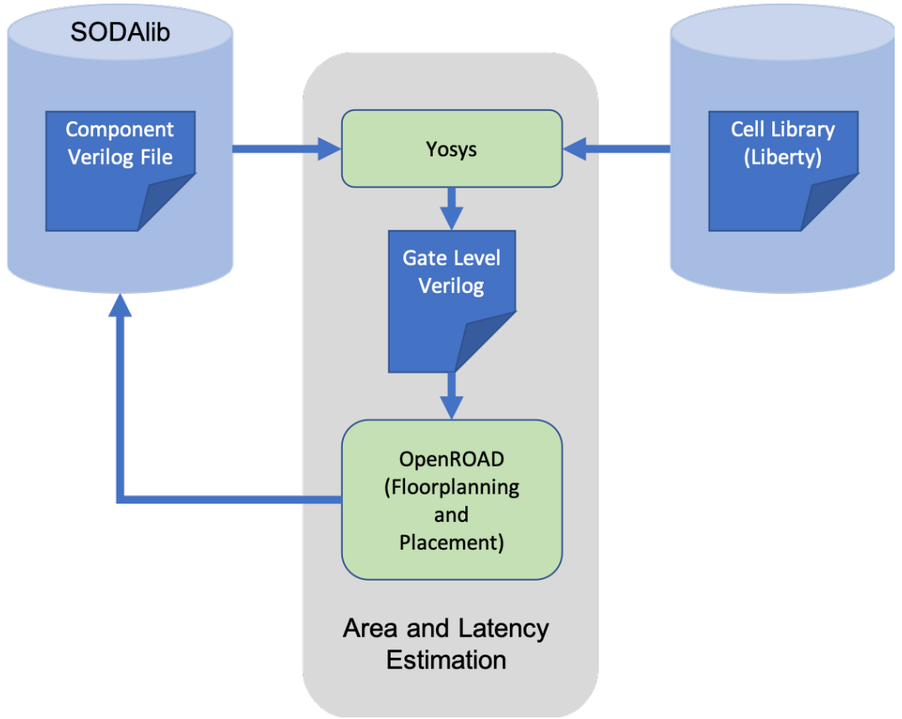
- New optimizations as additional compiler passes
- Design space exploration formulated as a compiler optimization problem

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

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

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



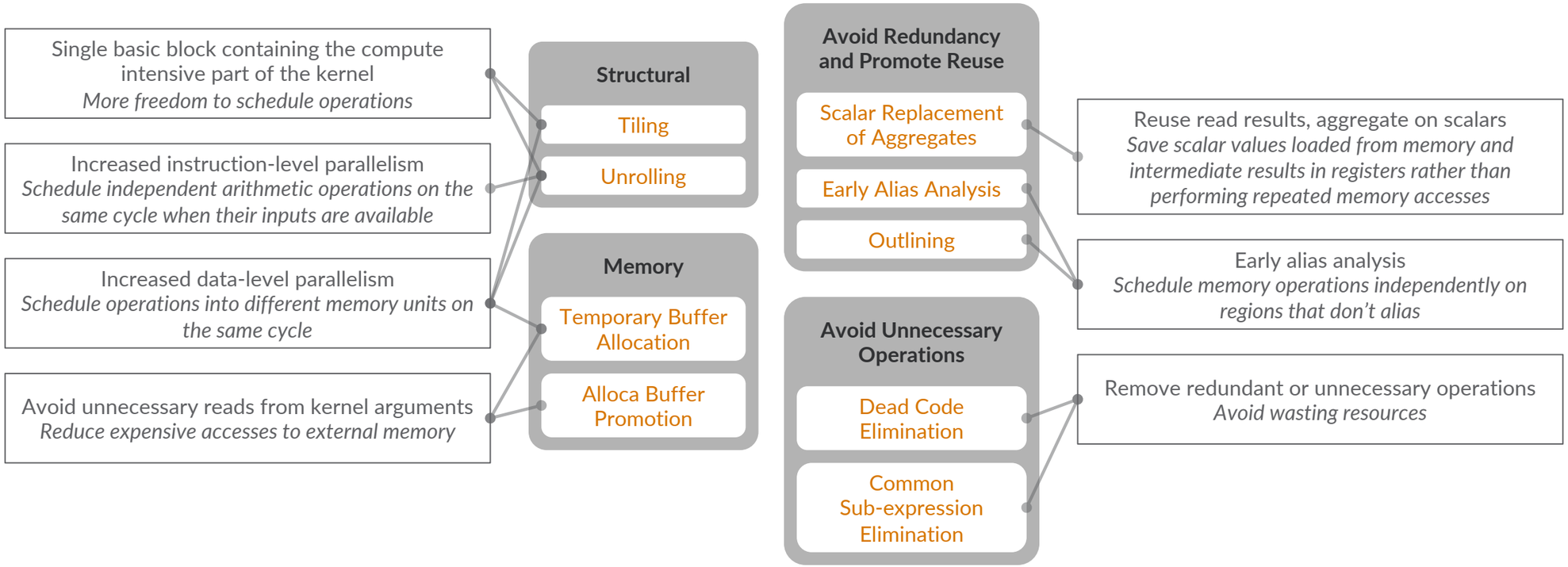
SODA characterization flow. The characterization flow can be extended to synthesize HLS generated designs, or used to estimate their area-latency-power profiles to drive the Design Space Exploration engine

OpenROAD

https://theopenroadproject.org

SODA-OPT Optimization Passes

- SODA-OPT implements optimizations as compiler passes



Pacific Northwest NATIONAL LABORATORY

PolyBench with ASAP 7nm

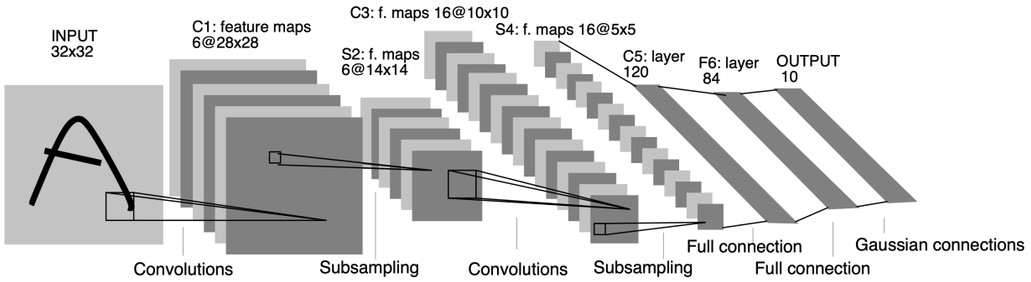
Kernel	Size	No optimizations				With optimizations				Trade-offs	
		Cycles	Max. freq. (MHz)	Area (μm^2)	Energy (nJ)	Cycles	Max. freq. (MHz)	Area (μm^2)	Energy (nJ)	Speedup	Area Overhead
atax	2	118	1164.28	1,658	1.63	38	986.65	2,539	1.13	3.11	1.53
	4	463	1384.20	1,970	8.20	63	1182.30	8,211	5.20	7.35	4.17
	8	1,819	1371.31	3,056	33.03	113	634.15	22,677	34.21	16.10	7.42
bicg	2	113	1778.91	1,605	1.26	24	1206.61	4,425	0.72	4.71	2.76
	4	458	1532.42	1,881	4.00	39	1098.86	12,015	3.87	11.74	6.39
	8	1,810	1314.11	2,974	34.98	78	400.86	30,744	21.01	23.21	10.34
gemm	2	161	1401.78	3,100	4.66	27	1411.24	5,067	1.09	5.96	1.63
	4	1,258	1286.68	4,597	37.74	51	782.38	21,089	16.36	24.67	4.59
	8	10,450	1451.15	2,590	143.30	139	–	–	–	75.18	–
gemver	2	246	1250.11	3,778	9.01	66	1374.54	5,832	2.78	3.73	1.54
	4	974	1031.52	7,835	68.27	91	652.62	18,595	43.50	10.70	2.37
	8	3,833	1121.75	7,547	292.83	141	–	–	–	27.18	–
gesummv	2	142	1554.66	2,336	2.25	35	1235.27	4,866	1.12	4.06	2.08
	4	514	1032.78	2,468	12.09	50	1128.52	10,431	5.10	10.28	4.23
	8	1,922	1279.50	3,614	44.16	101	638.13	25,487	52.07	19.03	7.05
mvt	2	114	1583.37	1,737	1.09	24	892.39	4,408	1.07	4.75	2.54
	4	450	1355.68	4,760	20.15	41	1125.35	13,366	3.59	10.98	2.81
	8	1,819	1216.38	3,544	60.56	81	559.19	32,757	16.08	22.46	9.24
three_mm	2	340	1155.26	3,577	9.42	42	1176.22	9,424	3.50	8.10	2.63
	4	2,719	1106.28	7,994	163.69	75	305.15	40,428	90.69	36.25	5.06
	8	22,130	1243.23	5,614	656.84	231	–	–	–	95.80	–
two_mm	2	274	1234.98	4,082	11.51	45	1337.67	6,915	0.78	6.09	1.69
	4	2,163	1019.49	7,063	128.57	75	422.71	34,326	20.94	28.84	4.86
	8	17,762	1281.06	4,251	508.85	–	–	–	–	–	–

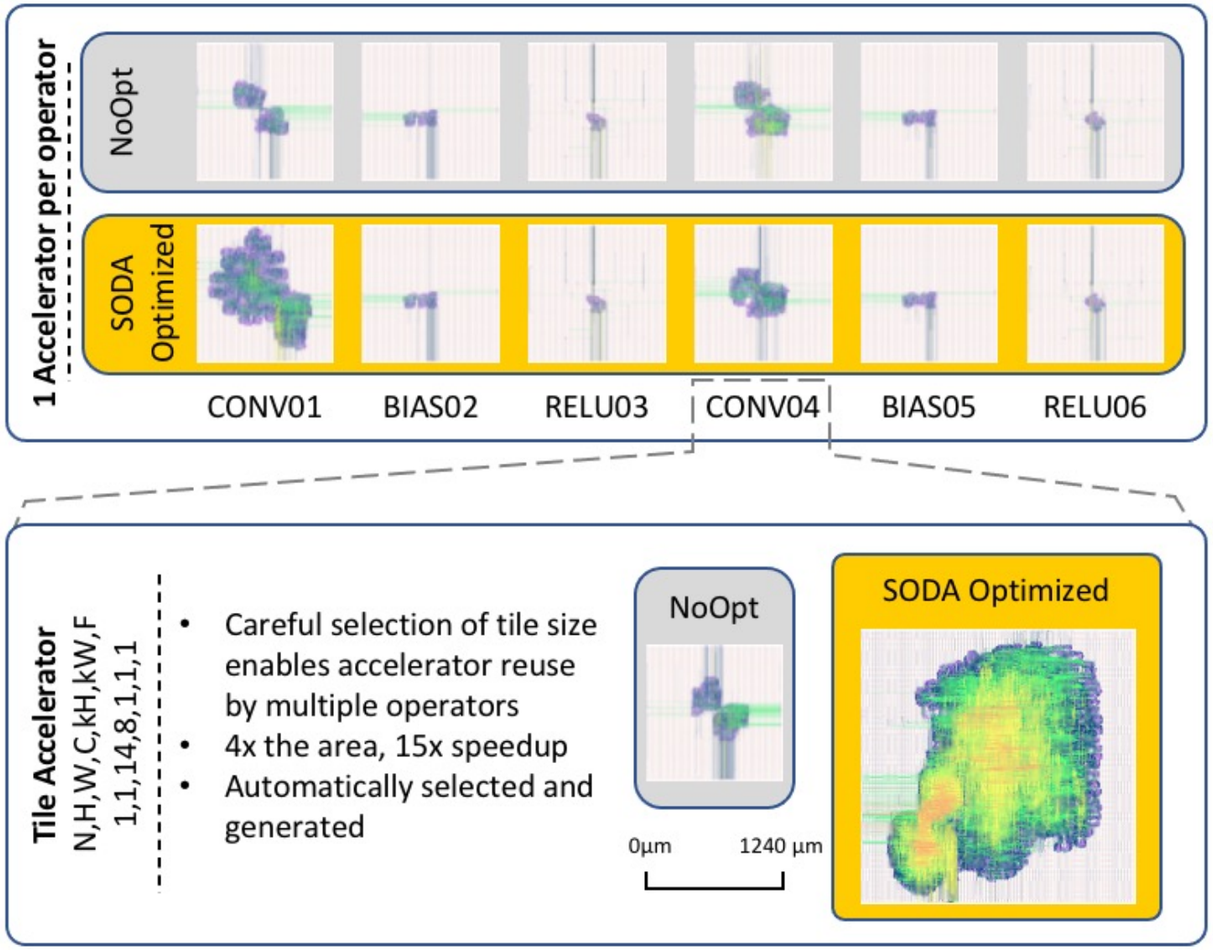
Table 1: PolyBench results with OpenROAD and the ASAP 7nm technology library

- Representative of **algorithmic patterns** in scientific computing or high-level data science frameworks
- Tensors of different **dimensions**
- Target frequency: ***1 GHz***
- Significant **speedups**, increased **energy efficiency** and comparatively small **area overheads**
- 1 GHz constraint met by ***all the non-optimized cases***
- Maximum frequency for the optimized cases **reduced for larger tensor sizes**
- ASIC designs **not generated** for a few optimized cases with larger tensor sizes because of **routing congestion**

From Python to optimized ASIC

LeNet architecture from the original paper





- LeNet example
- Each of the operator is synthesized to an **ASIC accelerator** using OpenROAD and FreePDK 45 nm
- SODA-Opt optimized accelerators are bigger, but also much **faster**

Pacific Northwest NATIONAL LABORATORY

LeNet with OpenPDK 45 nm

Kernel	No Optimizations			With optimizations			Trade-offs	
	Cycles	Area (μm^2)	Power eff. (GFLOPS/W)	Cycles	Area (μm^2)	Power eff. (GFLOPS/W)	Speedup	Area Overhead
CONV_01	10,262,618	29,073	4.43	4,627,982	124,255	2.68	2.22	4.27
BIAS_02	251,694	10,395	11.48	40,826	60,048	9.01	6.17	5.78
RELU_03	151,342	7,385	41.55	38,446	35,695	38.39	3.94	4.38
CONV_04	85,380,948	36,814	3.32	83,380,180	37,556	3.34	1.02	1.02
BIAS_05	62,932	10,409	11.00	10,222	60,007	8.41	6.16	5.76
RELU_06	37,844	7,464	41.75	9,620	35,950	37.04	3.93	4.82

Table 2: Evaluation of non optimized and optimized LeNet operators in ASIC technology (FreePDK 45 nm at 500 MHz)

- SODA-OPT's optimizations provide a speedup proportional to the increase in the area
- Power efficiency may be slightly reduced due to an increase in power consumption of the faster solutions

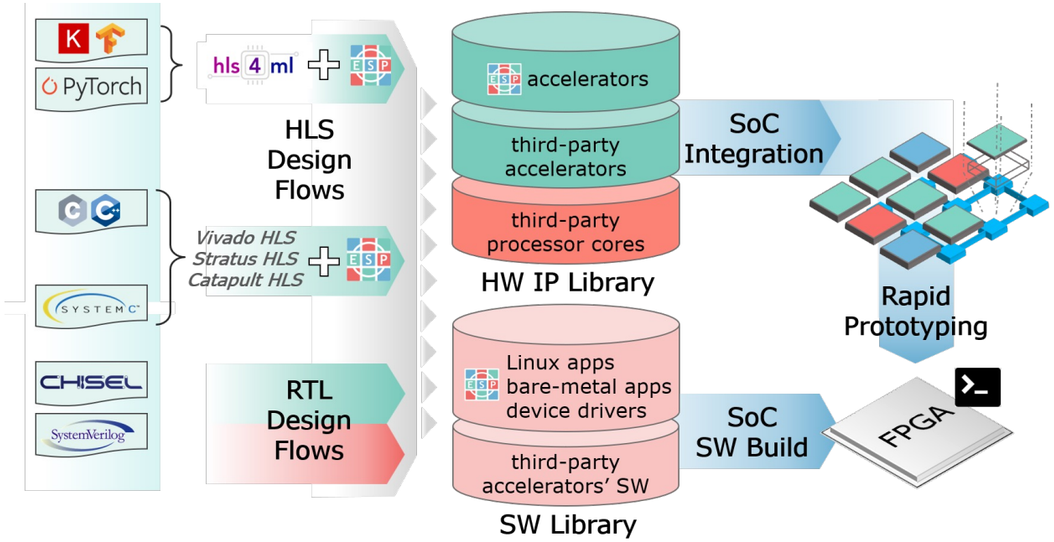
Pacific Northwest
NATIONAL LABORATORY

Research Opportunities: Open-Source Ecosystem

- SODA demonstrates how several Open-Source tools can seamlessly integrate
- SODA also provides initial support to commercial backends:
  - SODA-OPT generated LLVM IR can already be fed to Xilinx Vitis HLS
  - SODA also targets commercial ASIC logic synthesis tools
- Integration of proprietary tools, however, still is a significant challenges
- Significant opportunities in supporting:
  - Open-source intellectual property (IP) blocks as components in the resource libraries
  - Open-source system prototyping platforms
  - Open-source domain-specific FPGA generators to enable specialization starting from the high-level specifications

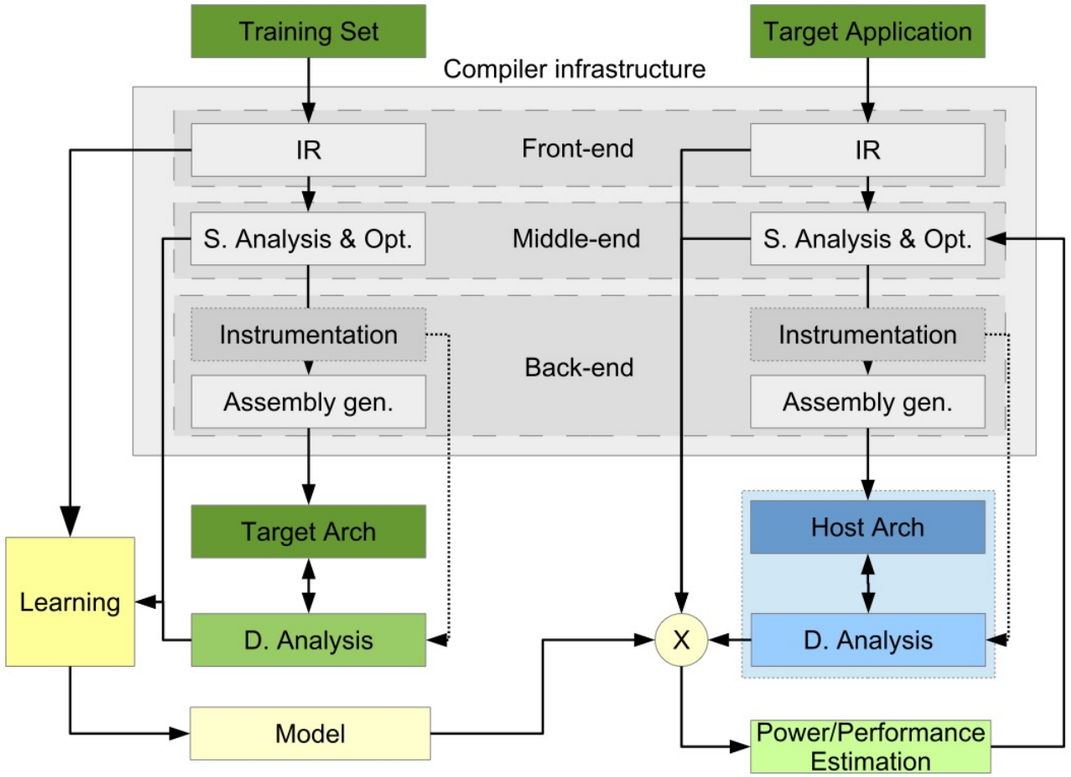
Research Opportunities: System-Level Design

- Integrating with open-source fast prototyping platforms: Columbia University Embedded Scalable Platforms (ESP)
- SODA-OPT
  - MLIR is naturally modular and hierarchical
  - Can lower to multiple targets, including runtimes
- Bambu
  - Provides a fully open-source HLS backend for ESP
- Enables end-to-end fast prototyping from algorithmic concept to system implementation



Research Opportunities: Profile Driven Synthesis

- A multi-level compiler infrastructure provides **static analysis**
- A compiler infrastructure provides opportunities to implement **dynamic analysis** through automated instrumentation and profiling
  - E.g., capturing data-dependent patterns and memory transactions
  - Information can be feed back to the hardware generation engine to facilitate exploration of the memory and the overall architecture design



[A. Tumeo: Architecture independent integrated early performance and energy estimation. IGSC 2017: 1-6]

Public Software Repositories

- SODA-Opt: https://gitlab.pnnl.gov/sodalite/soda-opt
- Panda-Bambu HLS: https://panda.dei.polimi.it (latest release 0.9.8)
- 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

PandA-Bambu HLS (v 0.9.8)

SODA Docker Image

SODA Tutorial: *DATE 2022*

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

- **The SODA Synthesizer provides a no-human-in-the-loop toolchain from algorithmic formulation to hardware implementation for complex workloads**

Pacific Northwest
NATIONAL LABORATORY

Thank you