# *LSOracle 2.0: Capabilities, Integration, and Performance*

***Scott Temple, Ashton Snelgrove, Walter Lau, Pierre-Emmanuel Gaillardon***

Department of Electrical and Computer Engineering – University of Utah

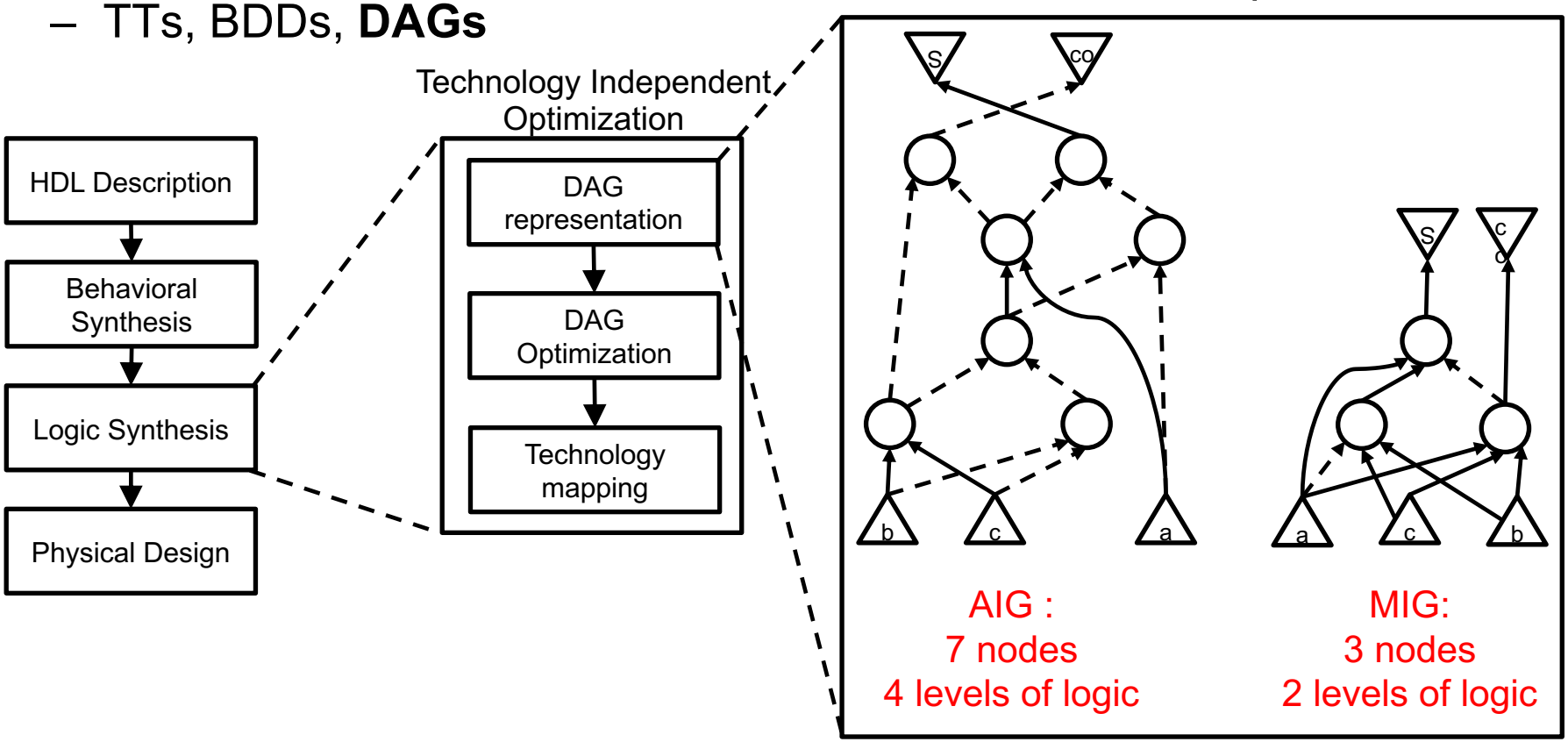

November 4th, 2021


# Outline

- Background
  - Logic Synthesis
  - Majority-Inverter Graphs
- LSOracle overview
  - Architecture
  - Benefits
  - New in Version 2.0
- Benchmarking
  - Runtime
  - QoR
- Ongoing work and future directions
- Conclusions

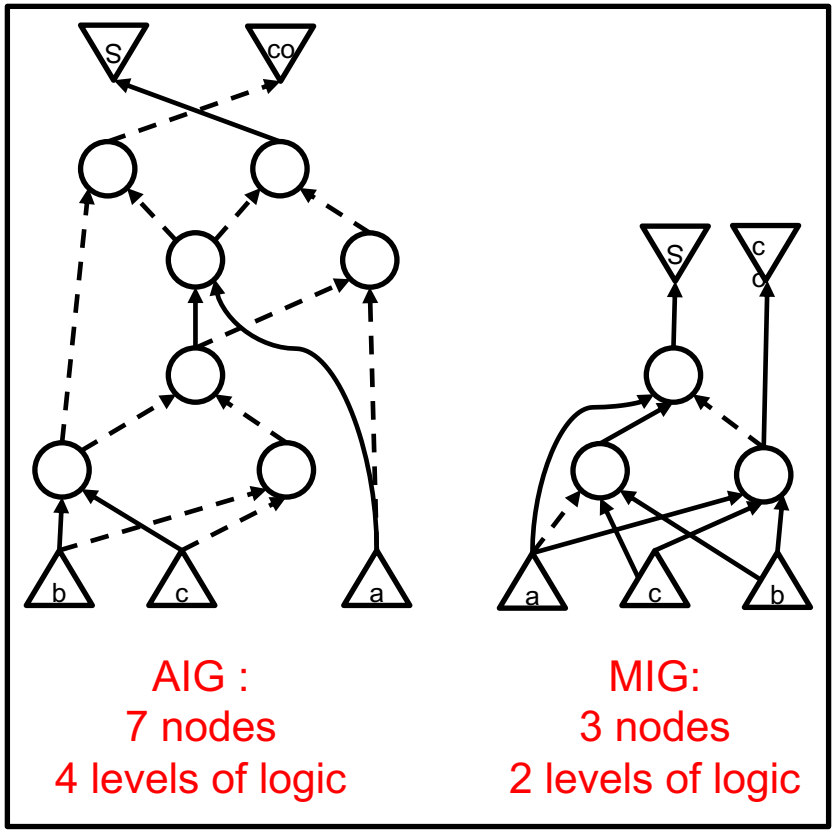
# Logic Synthesis in the EDA Flow

- EDA: Behavioral synthesis, logic synthesis, physical synthesis
- Logic Synthesis is at the forefront of EDA:
  - Strong impact on downstream tools
  - TTs, BDDs, **DAGs**



# DAGs in Logic Synthesis

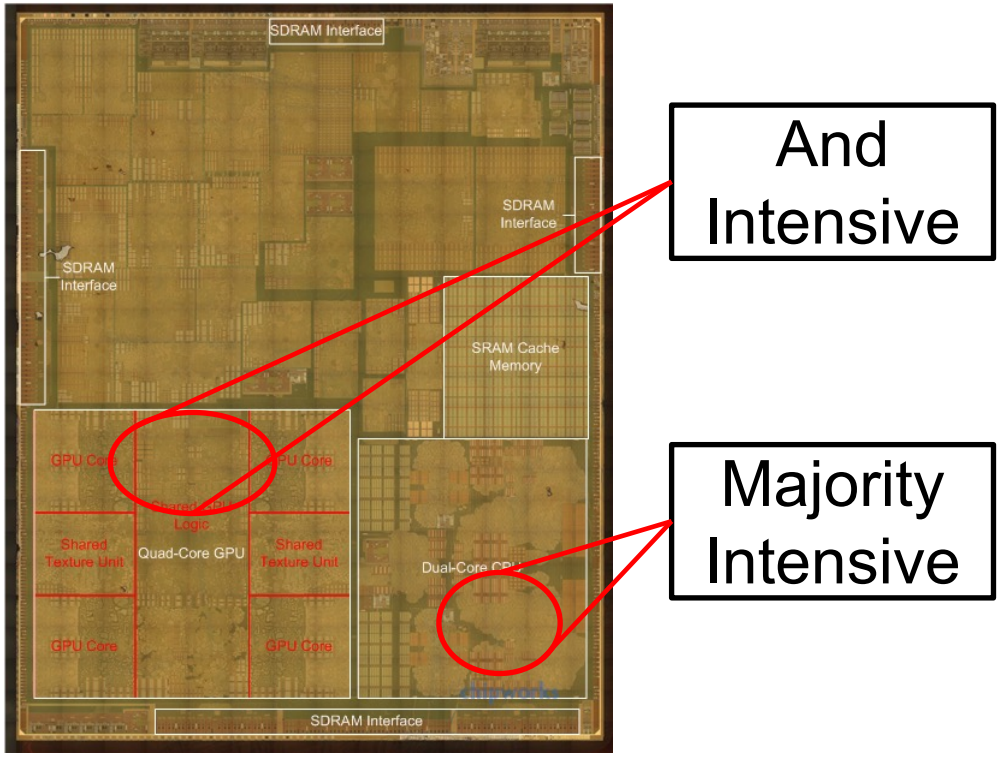
Full Adder Representation




- And-Inverter Graph (AIG)
  - Nodes: AND2
  - Edges: Regular or NOT
  - Excellent for random/control logic
  - Popular in open-source tools
- Majority-Inverter Graph (MIG)
  - Nodes: MAJ3
  - Edges: Regular or NOT
  - Superset of AIG
  - Excellent for arithmetic logic
- Others also possible (XAG, XMG)

# Why Mixed Logic Synthesis?

- No one approach is best for all types of circuit
- Real world designs typically have multiple types of logic
- A combination of optimization methods is desirable for large designs




And Intensive

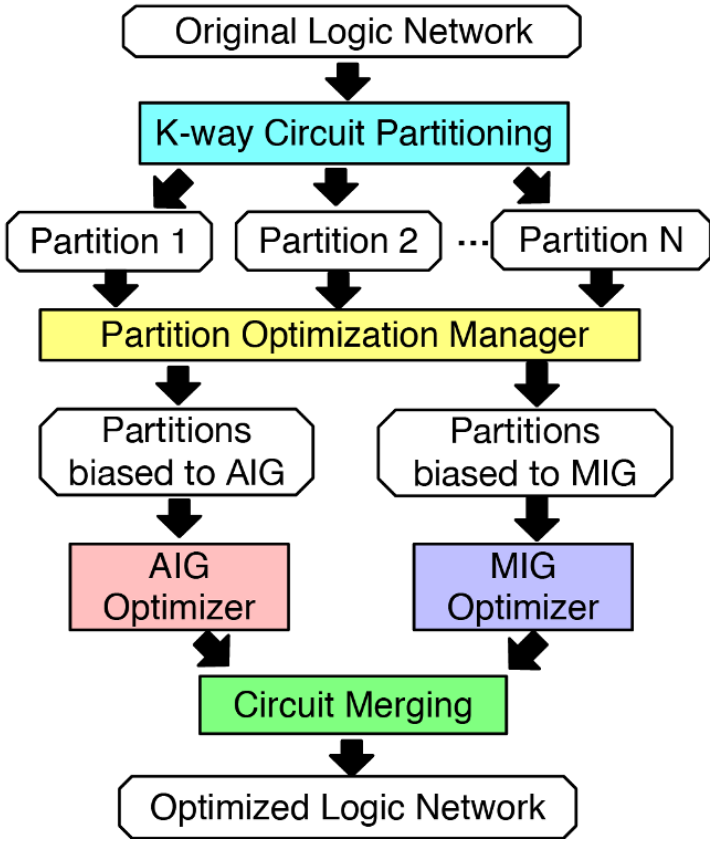
Majority Intensive

How to automatically select the right representation and optimizer for each part of a complex design?

Apple A8. Photo: TechInsights Inc.

# LSOracle

- **Key Idea**: Heterogeneous logic synthesis tailored to the underlying logic
- **Partition Circuit** into minimally dependent sections
- **Optimize partitions** selecting MIG or AIG for optimization
- **Merge logic** into a unified network and pass to techmapper and downstream tools

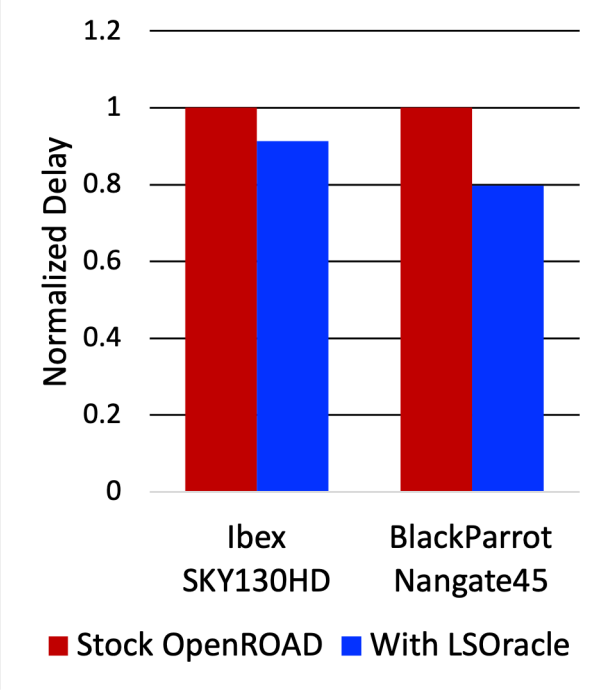


# Benefits of Mixed Synthesis

Significant reduction in delay after P&R across multiple designs and technology libraries

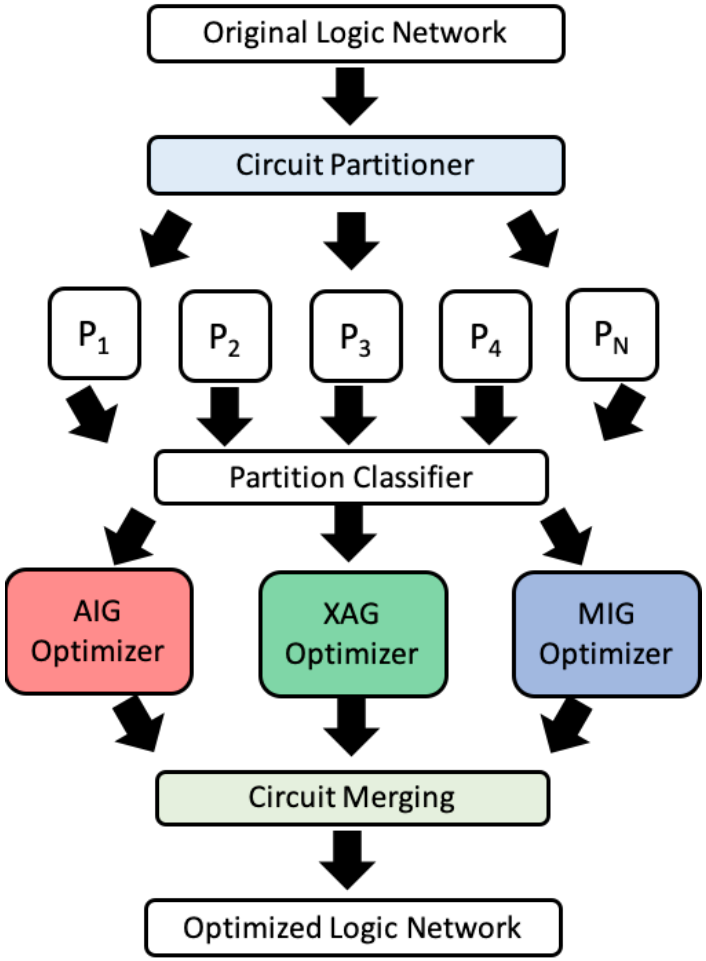
- Optimized with automatic AIG/MIG mixed synthesis
- Ibex
  - Small RISC V core
  - SKY130HD library
  - **9%** delay reduction
- BlackParrot
  - Linux capable RISC V core
  - Nangate45 library
  - **21%** delay reduction

Delay Improvements with Mixed Synthesis vs stock OpenROAD



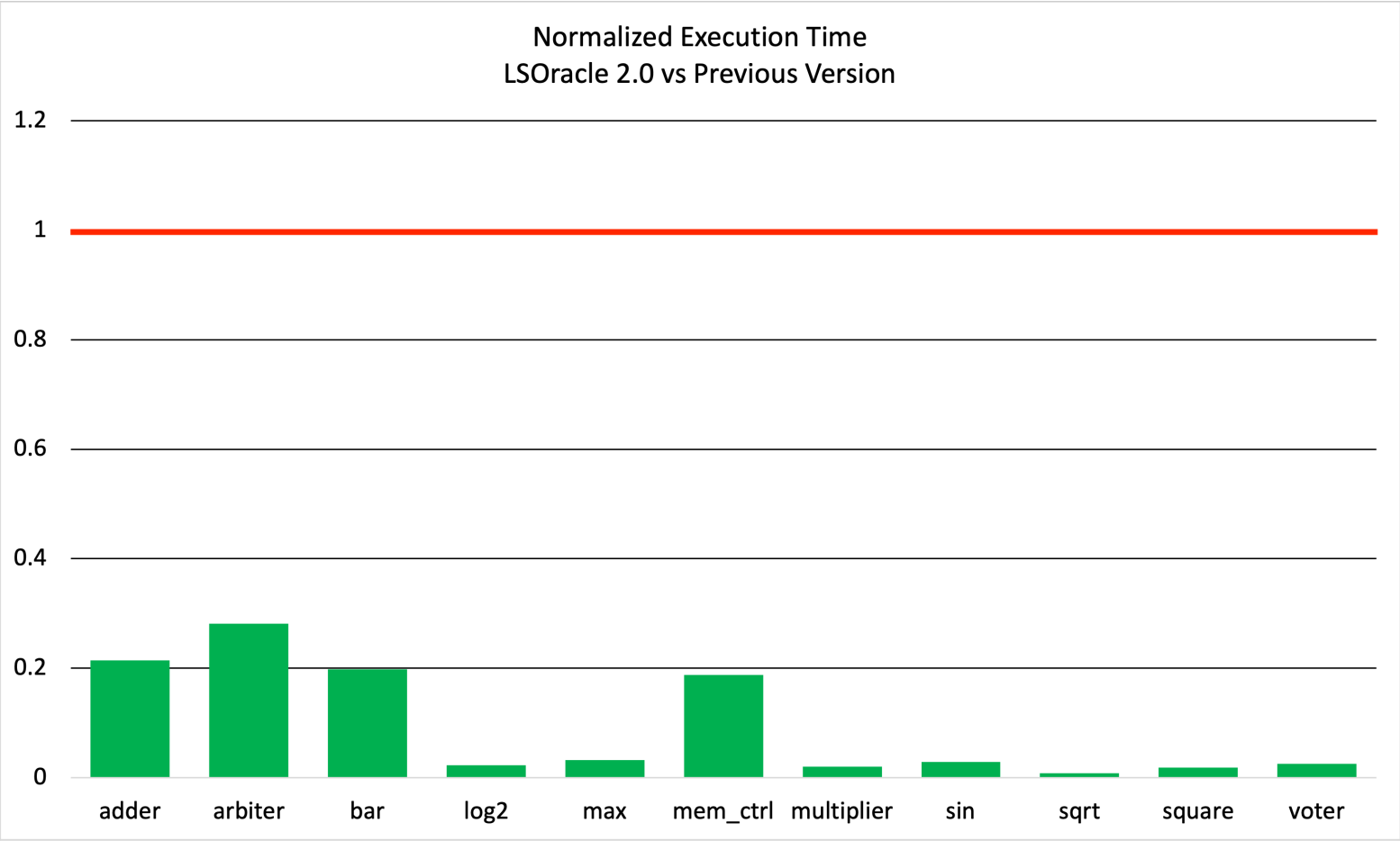
# New in Version 2

- Reduced runtime
- XAG and XMG support
- New optimization methods:
  - Functional reduction
  - Exact synthesis
- Additional optimization recipes
- Experimental features:
  - Native ASIC mapper
  - Equivalence checking
- Many under the hood improvements
- Included with OpenROAD



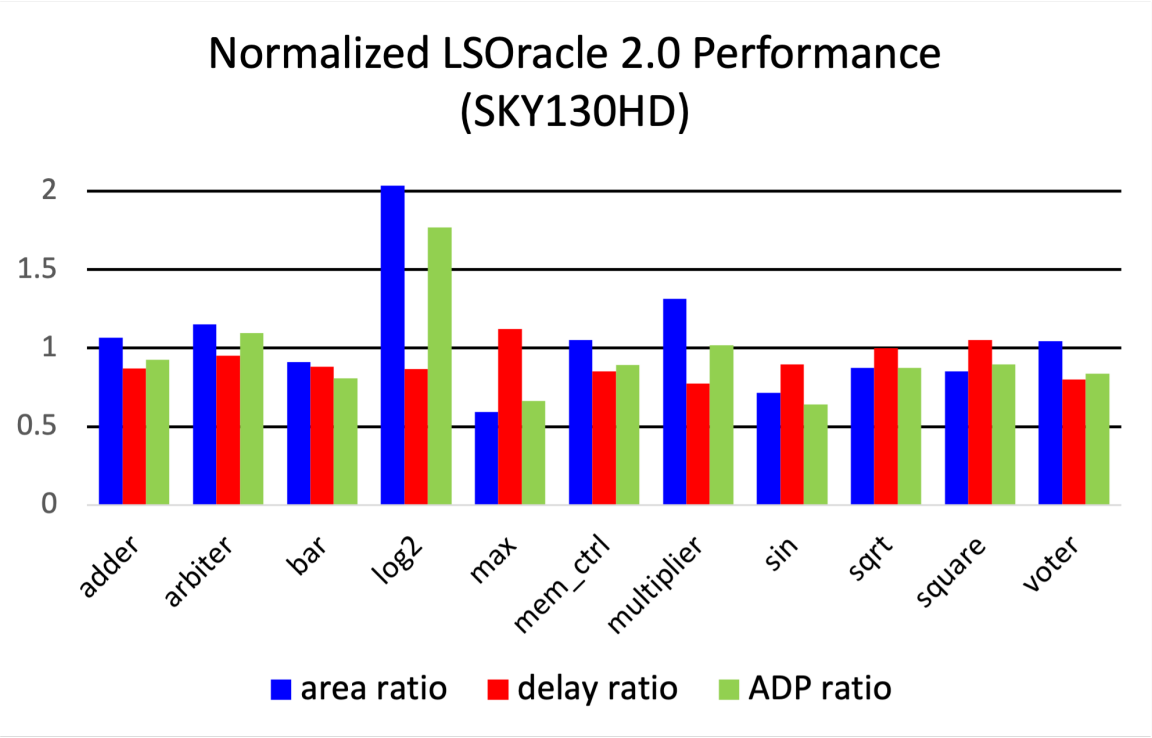
# Results: Runtime

On average 10× faster than previous release over EPFL benchmarks



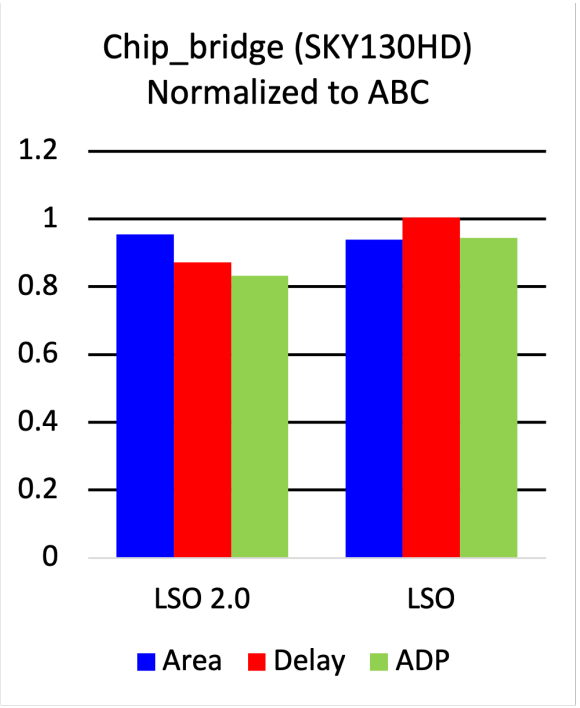
# PPA Results: EPFL Benchmarks

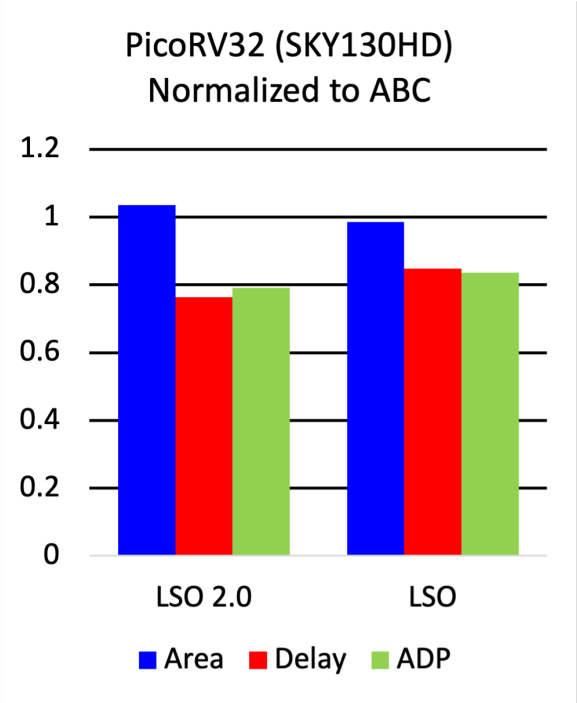
- Normalized against previous version
- AIG mode for control benchmarks, MIG for arithmetic
- Average
  - 8.5% delay reduction
  - 5.5% area increase



# PPA Results: Mixed Synthesis

- Tested mixed synthesis mode on two circuits
- Chip bridge (OPDB): 13% delay reduction
- PicoRV: 10% delay reduction

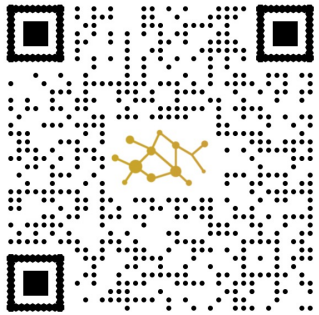


# Ongoing Work & Getting LSOracle

- Current development is focused on:
  - Timing-driven synthesis
  - Native technology mapping
  - Applying machine learning

- LSOracle is MIT licensed and is available:
  - On Github: https://github.com/lnis-uofu/LSOracle
  - On Docker Hub: https://hub.docker.com/orgs/lnis
  - On Launchpad: https://launchpad.net/lsoracle
  - As part of OpenROAD

# Thank you for your attention

**Laboratory for NanoIntegrated Systems**
Department of Electrical and Computer Engineering
MEB building – University of Utah – Salt Lake City – UT – USA