Introduction to Machine Learning for Computer-Aided Design (CAD)

EEC S 151/251A
Josh Kang (advised by John Wawrzynek)
Mar 21 2023

Berkeley
UNIVERSITY OF CALIFORNIA

Agenda

- Overview of Recent Trends in ML for CAD/EDA
    - Deeper Dive: ML-Driven Logic Synthesis Optimization
- Challenges in ML for CAD
- Research @ Berkeley on ML-CAD

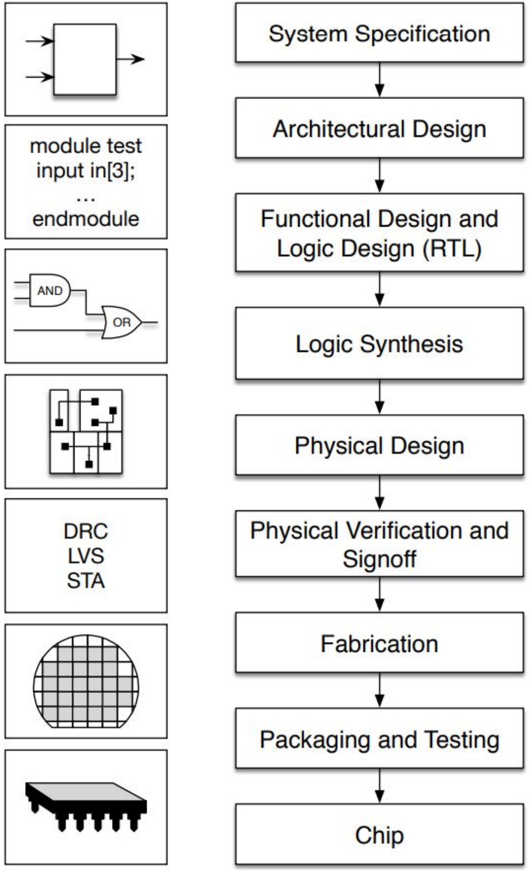
1

Overview of
Recent ML-CAD Research

Berkeley
UNIVERSITY OF CALIFORNIA

ML for Various Stages of Digital IC Design

- Active research on applying ML (notably Deep Learning) to each stage of EDA
- Each stage can have multiple tasks to target:
  - e.g. in P&R:
    - Predict routing congestion
    - Predict routing result
  - Full or partial automation
  - Improve optimization of quality of results (QoR)
- But also, ML for AMS Design



```
module test
input in[3];
...
endmodule
```

DRC
LVS
STA

System Specification

Architectural Design

Functional Design and Logic Design (RTL)

Logic Synthesis

Physical Design

Physical Verification and Signoff

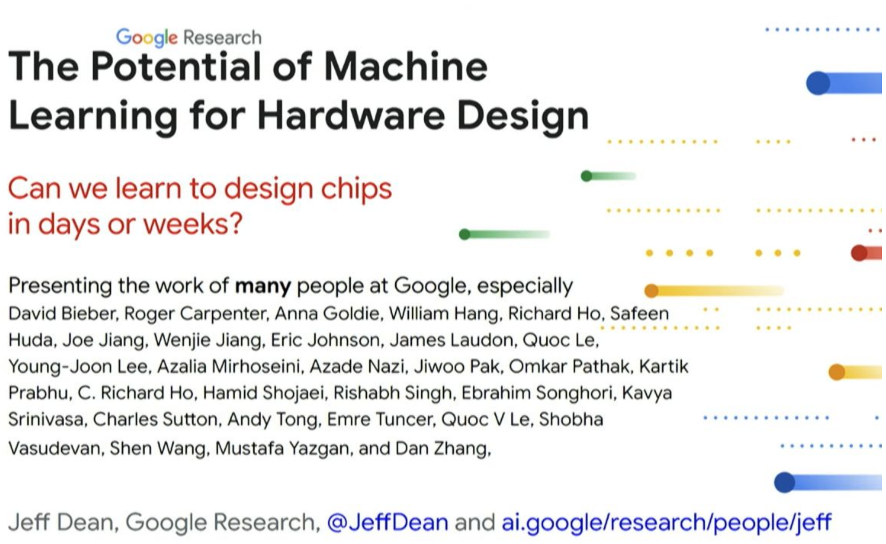
Fabrication

Packaging and Testing

Chip

Huang, Guyue, et al. "Machine learning for electronic design automation: A survey." *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 26.5 (2021)

Recent Trends: ML-CAD @ Industry



Google Research

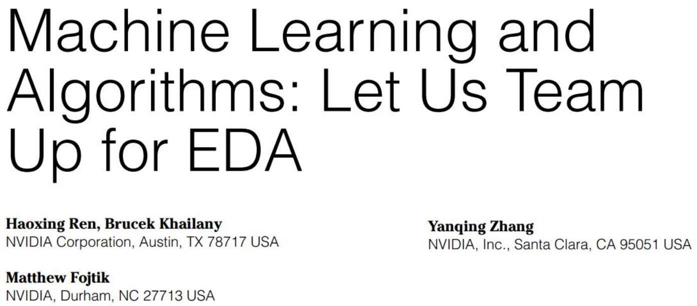
The Potential of Machine Learning for Hardware Design

Can we learn to design chips in days or weeks?

Presenting the work of **many** people at Google, especially David Bieber, Roger Carpenter, Anna Goldie, William Hang, Richard Ho, Safeen Huda, Joe Jiang, Wenjie Jiang, Eric Johnson, James Laudon, Quoc Le, Young-Joon Lee, Azalia Mirhoseini, Azade Nazi, Jiwoo Pak, Omkar Pathak, Kartik Prabhu, C. Richard Ho, Hamid Shojaei, Rishabh Singh, Ebrahim Songhori, Kavya Srinivasa, Charles Sutton, Andy Tong, Emre Tuncer, Quoc V Le, Shobha Vasudevan, Shen Wang, Mustafa Yazgan, and Dan Zhang,

Jeff Dean, Google Research, @JeffDean and ai.google/research/people/jeff

Google @DAC 2022



Machine Learning and Algorithms: Let Us Team Up for EDA

Haoxing Ren, Brucek Khailany
NVIDIA Corporation, Austin, TX 78717 USA

Yanqing Zhang
NVIDIA, Inc., Santa Clara, CA 95051 USA

Matthew Fojtik
NVIDIA, Durham, NC 27713 USA

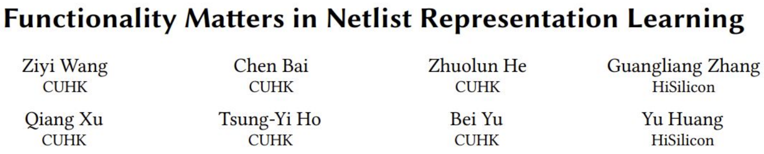
The Intelligence to Design Intelligent Machines
Ben Gu
Vice President R&D, Multiphysics System Analysis
Cadence Design Systems, Inc.
DesignCon 2023
JAN. 31 – FEB. 2, 2023
#DesignCon

Cadence

@DesignCon

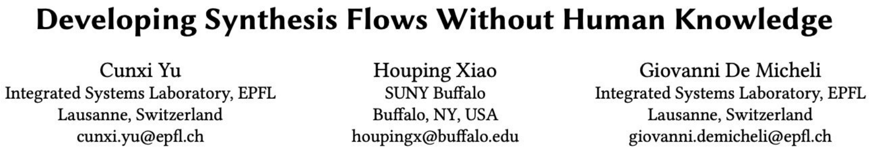
2023

Recent Trends: ML-CAD @ Academia



Functionality Matters in Netlist Representation Learning

Ziyi Wang CUHK	Chen Bai CUHK	Zhuolun He CUHK	Guangliang Zhang HiSilicon
Qiang Xu CUHK	Tsung-Yi Ho CUHK	Bei Yu CUHK	Yu Huang HiSilicon

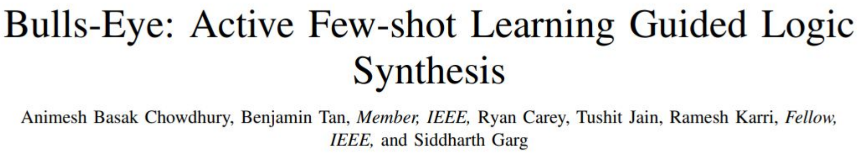


Developing Synthesis Flows Without Human Knowledge

Cunxi Yu
Integrated Systems Laboratory, EPFL
Lausanne, Switzerland
cunxi.yu@epfl.ch

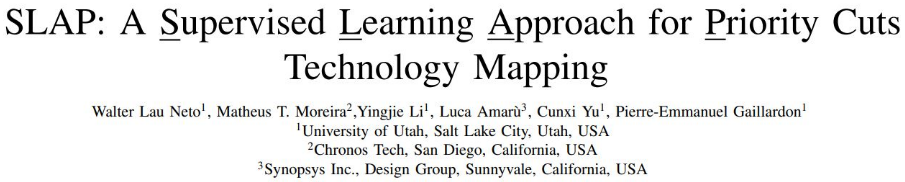
Houping Xiao
SUNY Buffalo
Buffalo, NY, USA
houpingx@buffalo.edu

Giovanni De Micheli
Integrated Systems Laboratory, EPFL
Lausanne, Switzerland
giovanni.demicheli@epfl.ch



Bulls-Eye: Active Few-shot Learning Guided Logic Synthesis

Animesh Basak Chowdhury, Benjamin Tan, *Member, IEEE,* Ryan Carey, Tushit Jain, Ramesh Karri, *Fellow, IEEE,* and Siddharth Garg



SLAP: A Supervised Learning Approach for Priority Cuts Technology Mapping

Walter Lau Neto[1], Matheus T. Moreira[2],Yingjie Li[1], Luca Amarù[3], Cunxi Yu[1], Pierre-Emmanuel Gaillardon[1]
[1]University of Utah, Salt Lake City, Utah, USA
[2]Chronos Tech, San Diego, California, USA
[3]Synopsys Inc., Design Group, Sunnyvale, California, USA

- Major CAD conferences (DAC, ICCAD, DATE, etc.) are featuring increasing numbers of research on ML-CAD
- Specific workshop conference: ACM/IEEE Workshop on MLCAD

Deeper Dive: ML for Logic Synthesis

- Early Stage in IC Design: unoptimized layout from synthesis will hurt end PPA

- “Re-inventing” Logic Synthesis tools with ML methods is impractical / too complex

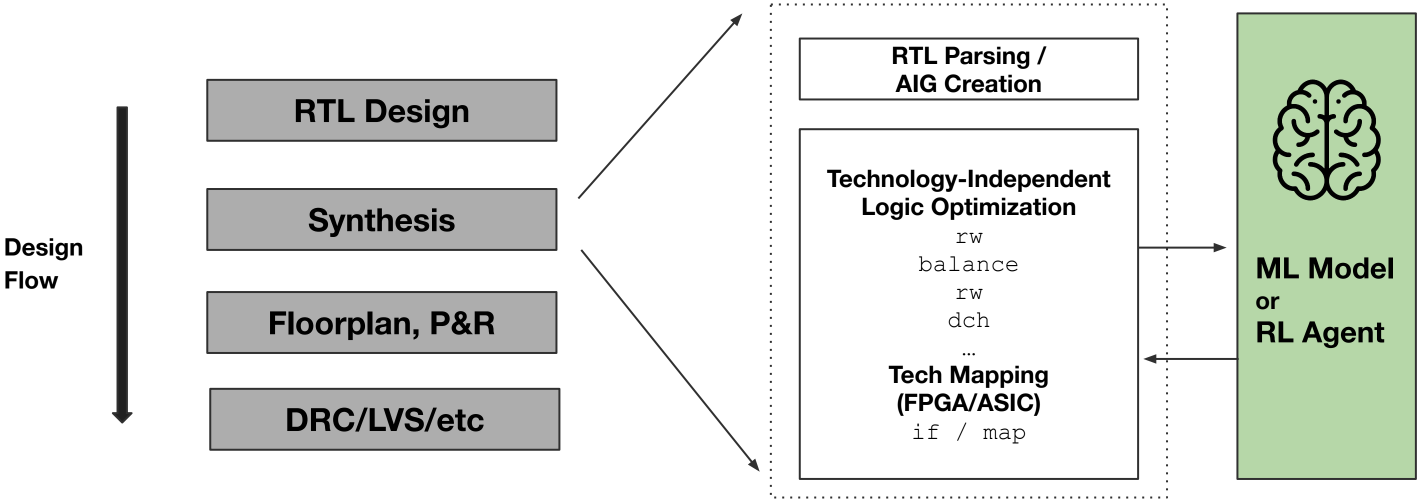
- Instead, **replace heuristic-based decisions** made during logic synthesis with ML
  - Order sequence of Boolean optimizations:

    Flowtune (Multi-Arm Bandit), **BOiLS** (GP Bayes-Opt), DeepRL-based works, etc.
  - Predict post-synthesis PPA: SNS, etc.
  - Choice of optimizer / logic representation (AIG/MIG): LSOracle, etc.

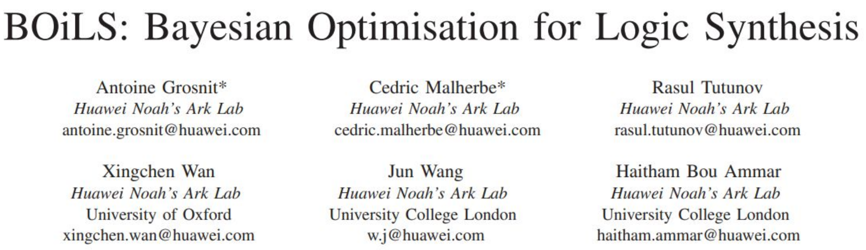
Deeper Dive: ML for Logic Synthesis

- Instead, **replace heuristic-based decisions** made during logic synthesis with ML
  - Order sequence of Boolean optimizations:

    Flowtune (Multi-Arm Bandit), **BOiLS** (GP Bayes-Opt), DeepRL-based works, etc.



Deeper Dive: ML for Logic Synthesis



BOiLS: Bayesian Optimisation for Logic Synthesis

Antoine Grosnit*
Huawei Noah's Ark Lab
antoine.grosnit@huawei.com

Cedric Malherbe*
Huawei Noah's Ark Lab
cedric.malherbe@huawei.com

Rasul Tutunov
Huawei Noah's Ark Lab
rasul.tutunov@huawei.com

Xingchen Wan
Huawei Noah's Ark Lab
University of Oxford
xingchen.wan@huawei.com

Jun Wang
Huawei Noah's Ark Lab
University College London
w.j@huawei.com

Haitham Bou Ammar
Huawei Noah's Ark Lab
University College London
haitham.ammar@huawei.com

$$\mathtt{seq}^\star \in \underset{\mathtt{seq}\in\mathtt{Alg}^K}{\arg\min}\ \mathrm{QoR}_{\mathcal{C}}(\mathtt{seq}) \equiv \underbrace{\arg\max\ \text{-}\mathrm{QoR}_{\mathcal{C}}(\mathtt{seq})}_{\text{This paper's focus}}$$

$$\mathrm{QoR}_{\mathcal{C}}(\mathtt{seq}) = \frac{\mathrm{Area}_{\mathcal{C}}(\mathtt{seq})}{\mathrm{Area}_{\mathcal{C}}(\mathtt{ref})} + \frac{\mathrm{Delay}_{\mathcal{C}}(\mathtt{seq})}{\mathrm{Delay}_{\mathcal{C}}(\mathtt{ref})}$$

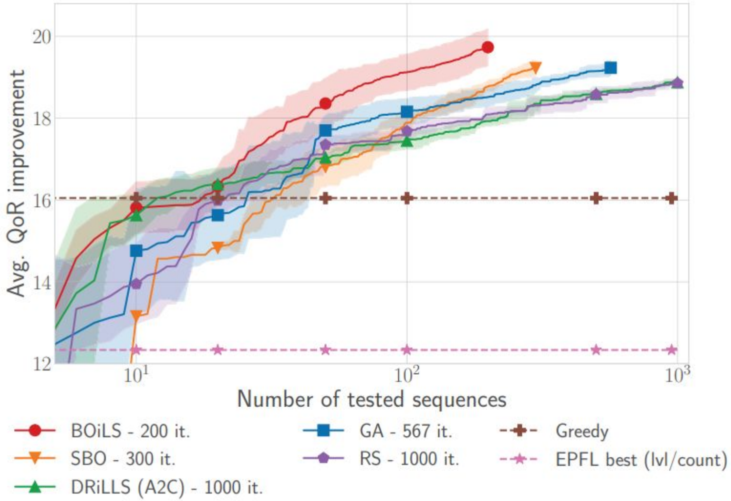


Fig. 1. Average QoR results over 10 EPFL circuits to recover BOiLS' QoR values after only *200* trials.

Grosnit, Antoine, et al. "Boils: Bayesian optimisation for logic synthesis." *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2022.

Deeper Dive: ML for Logic Synthesis

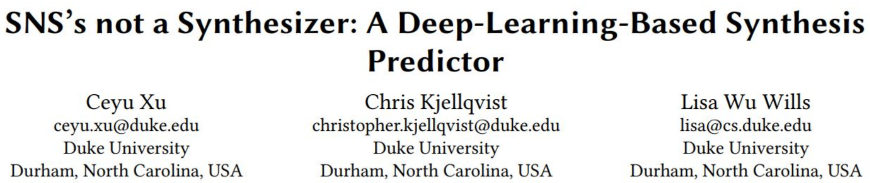
- Early Stage in IC Design: unoptimized layout from synthesis will hurt end PPA

- “Re-inventing” Logic Synthesis tools with ML methods is impractical / too complex

- Instead, **replace heuristic-based decisions** made during logic synthesis with ML
    - Order sequence of Boolean optimizations:

      Flowtune (Multi-Arm Bandit), BOiLS (GP Bayes-Opt), DeepRL-based works, etc.
    - Predict post-synthesis PPA: **SNS**, etc.
    - Choice of optimizer / logic representation (AIG/MIG): LSOracle, etc.

Deeper Dive: ML for Logic Synthesis



SNS's not a Synthesizer: A Deep-Learning-Based Synthesis Predictor

Ceyu Xu
ceyu.xu@duke.edu
Duke University
Durham, North Carolina, USA

Chris Kjellqvist
christopher.kjellqvist@duke.edu
Duke University
Durham, North Carolina, USA

Lisa Wu Wills
lisa@cs.duke.edu
Duke University
Durham, North Carolina, USA

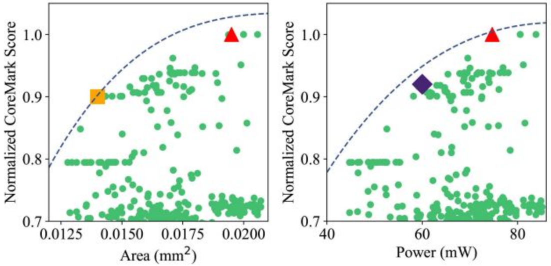
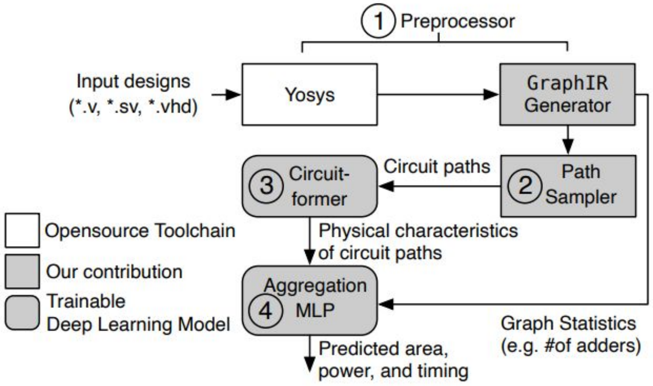
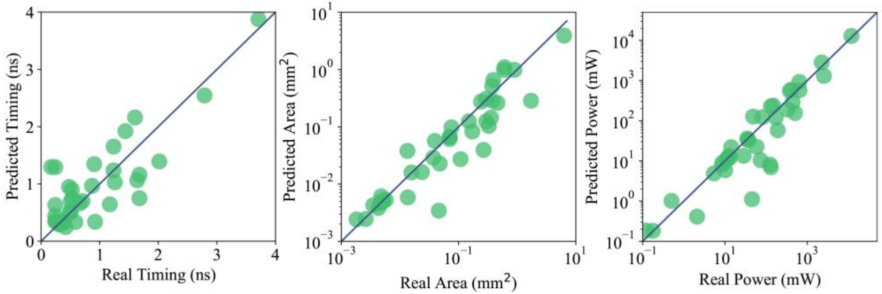


Figure 8: Boom DSE Result: The `HighPerf` design is plotted using a triangle shape, the `PowerEff` design is plotted using a diamond shape, and the `AreaEff` design is plotted using a square shape. The coremark scores are linearly normalized such that the fastest core in our design has a score of one.





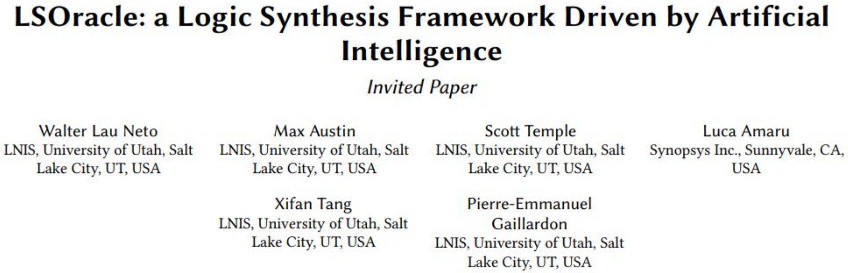
Xu, Ceyu, Chris Kjellqvist, and Lisa Wu Wills. "SNS's not a synthesizer: a deep-learning-based synthesis predictor." *Proceedings of the 49th Annual International Symposium on Computer Architecture*. 2022.

Deeper Dive: ML for Logic Synthesis

- Early Stage in IC Design: unoptimized layout from synthesis will hurt end PPA
- “Re-inventing” Logic Synthesis tools with ML methods is impractical / too complex
- Instead, **replace heuristic-based decisions** made during logic synthesis with ML
    - Order sequence of Boolean optimizations:

      Flowtune (Multi-Arm Bandit), BOiLS (GP Bayes-Opt), DeepRL-based works, etc.
    - Predict post-synthesis PPA: SNS, etc.
    - Choice of optimizer / logic representation (AIG/MIG): **LSOracle**, etc.

Deeper Dive: ML for Logic Synthesis



LSOracle: a Logic Synthesis Framework Driven by Artificial Intelligence

Invited Paper

Walter Lau Neto
LNIS, University of Utah, Salt Lake City, UT, USA

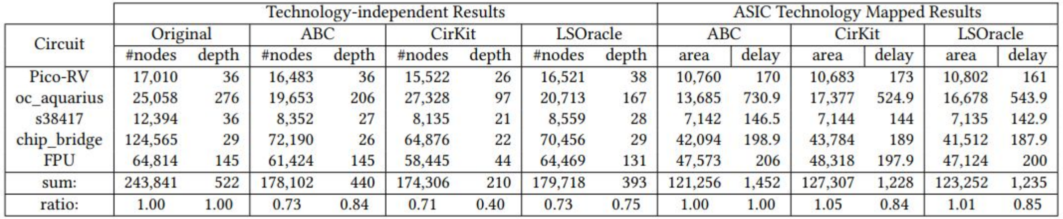
Max Austin
LNIS, University of Utah, Salt Lake City, UT, USA

Scott Temple
LNIS, University of Utah, Salt Lake City, UT, USA

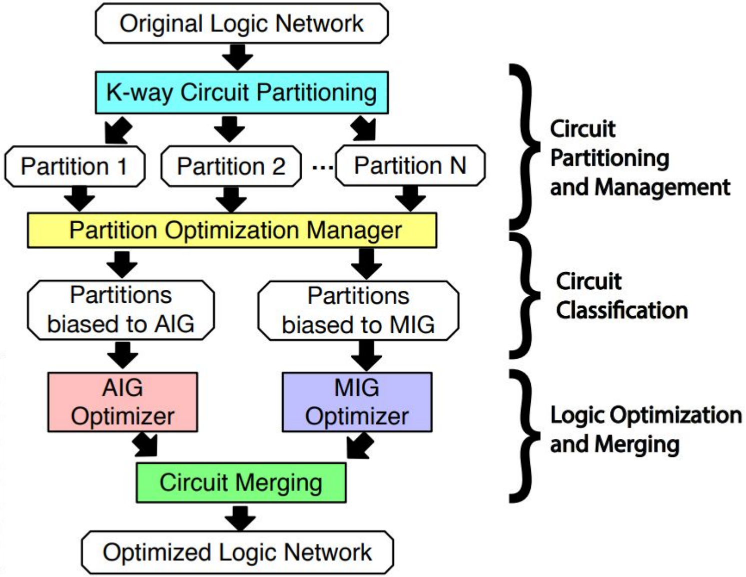
Luca Amaru
Synopsys Inc., Sunnyvale, CA, USA

Xifan Tang
LNIS, University of Utah, Salt Lake City, UT, USA

Pierre-Emmanuel Gaillardon
LNIS, University of Utah, Salt Lake City, UT, USA



	Technology-independent Results								ASIC Technology Mapped Results					
Circuit	Original		ABC		CirKit		LSOracle		ABC		CirKit		LSOracle	
	#nodes	depth	#nodes	depth	#nodes	depth	#nodes	depth	area	delay	area	delay	area	delay
Pico-RV	17,010	36	16,483	36	15,522	26	16,521	38	10,760	170	10,683	173	10,802	161
oc_aquarius	25,058	276	19,653	206	27,328	97	20,713	167	13,685	730.9	17,377	524.9	16,678	543.9
s38417	12,394	36	8,352	27	8,135	21	8,559	28	7,142	146.5	7,144	144	7,135	142.9
chip_bridge	124,565	29	72,190	26	64,876	22	70,456	29	42,094	198.9	43,784	189	41,512	187.9
FPU	64,814	145	61,424	145	58,445	44	64,469	131	47,573	206	48,318	197.9	47,124	200
sum:	243,841	522	178,102	440	174,306	210	179,718	393	121,256	1,452	127,307	1,228	123,252	1,235
ratio:	1.00	1.00	0.73	0.84	0.71	0.40	0.73	0.75	1.00	1.00	1.05	0.84	1.01	0.85



Neto, Walter Lau, et al. "LSOracle: A logic synthesis framework driven by artificial intelligence." *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2019.

2

Challenges in ML-CAD

Berkeley
UNIVERSITY OF CALIFORNIA

When is ML/DL Successful?

- **Abundance of Data** and **Availability of Compute**
- Suppose compute is not a problem (enough GPUs for the model/dataset size)
- Example Datasets in Computer Vision: Image Classification
  - CIFAR-10: 6000 images per class, total 60K images
  - ImageNet: ~1000 images per synonym set, 100K+ synonym sets
- Example Datasets in NLP: Language Modeling
  - WikiText-103: 100M tokens from > 28K selected Wikipedia articles

Challenges in ML-CAD

- **Lack of Publicly Available Digital Circuit Datasets**
  - Relative shortage of Open-Source RTL
    - < 50 substantial & full-developed open-source RTL projects/suites
    - “Standardization” of data format a huge burden
      - e.g. if use Graph Neural Networks (GNNs), which netlist representation (AST, Gate-level, etc.) to use and how to extract graphs from different HDLs?
  - Most work employ **Supervised Learning**:

    labeling cost often incurs 1+ run of commercial/gold-standard EDA flow
    - For realistic workloads, could take order of minutes/hours/days per datapoint
  - Similar for Deep RL: true reward evaluation incurs significant overhead
  - **Data and label efficiency is critical to success**

Challenges in ML-CAD

- **Implications of the Lack of Established, Large (Supervised) Datasets**
  - (1) Model cannot scale and **difficult to generalize across multiple circuit designs**
    - Greatly limits practical usage of ML in EDA workflows
  - (2) Many works are not evaluated against realistic designs and/or

    sufficient numbers of such designs
  - (3) Substantial data collection (per-task) + ML training cost:

    could have used run traditional toolflows longer for the same amount of time
- **Ideas to Overcome these Limitations**
  - Data Augmentation and Transfer Learning
    - Train with easier-to-obtain labels; transfer to actual task
    - Domain adaptation to different circuit designs

      (Mirhoseini et al. “Chip Placement with Deep Reinforcement Learning” 2020, etc.)

Challenges in ML-CAD

- **Ideas to Overcome this Limitations**
  - Data Augmentation Example:
    - **SNS's not a Synthesizer (Xu et al. ISCA 2022)**
    - Sample FF-to-FF paths from RTL graph
    - Model trained to predict PPA of individual paths;

      Generalize to predicting full circuit PPA

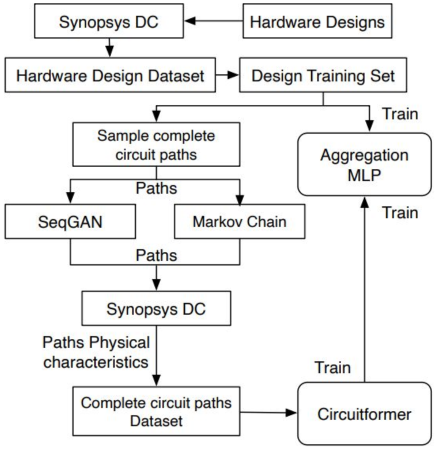


Figure 4: SNS Training Flow

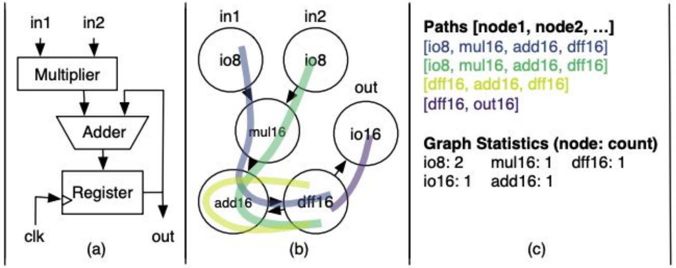
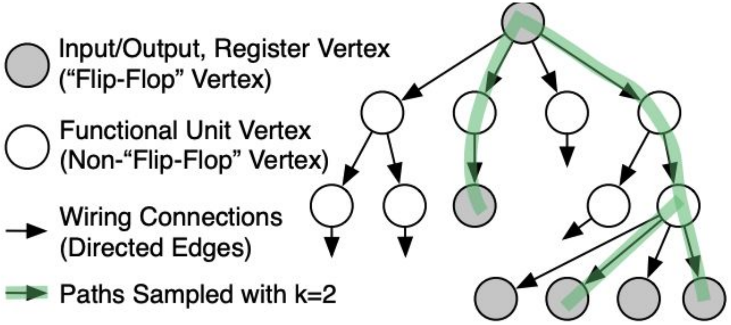


Figure 2: Transformation from Source Circuit to GraphIR Representation with Path Information and Graph Statistics



Input/Output, Register Vertex ("Flip-Flop" Vertex)
Functional Unit Vertex (Non-"Flip-Flop" Vertex)
Wiring Connections (Directed Edges)
Paths Sampled with k=2

3

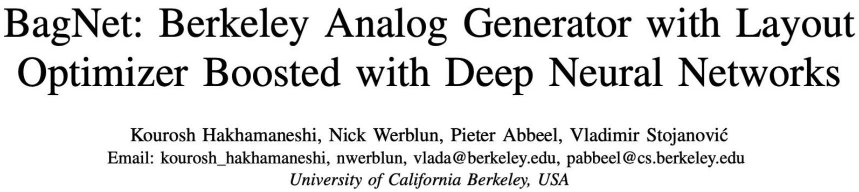
Research on ML-CAD @ Berkeley

Berkeley
UNIVERSITY OF CALIFORNIA

Berkeley Research on ML-CAD

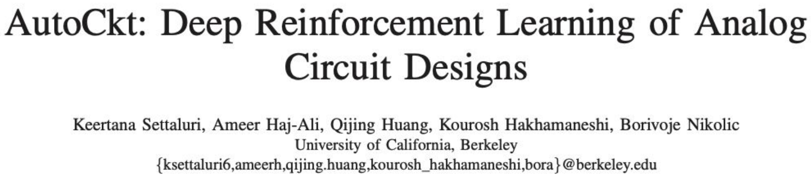
- **ML for Analog-Mixed Signal (AMS) Circuit Design:**

  Previous students at Stojanovic and Nikolic Groups



BagNet: Berkeley Analog Generator with Layout Optimizer Boosted with Deep Neural Networks

Kourosh Hakhamaneshi, Nick Werblun, Pieter Abbeel, Vladimir Stojanović
Email: kourosh_hakhamaneshi, nwerblun, vlada@berkeley.edu, pabbeel@cs.berkeley.edu
University of California Berkeley, USA



AutoCkt: Deep Reinforcement Learning of Analog Circuit Designs

Keertana Settaluri, Ameer Haj-Ali, Qijing Huang, Kourosh Hakhamaneshi, Borivoje Nikolic
University of California, Berkeley
{ksettaluri6,ameerh,qijing.huang,kourosh_hakhamaneshi,bora}@berkeley.edu

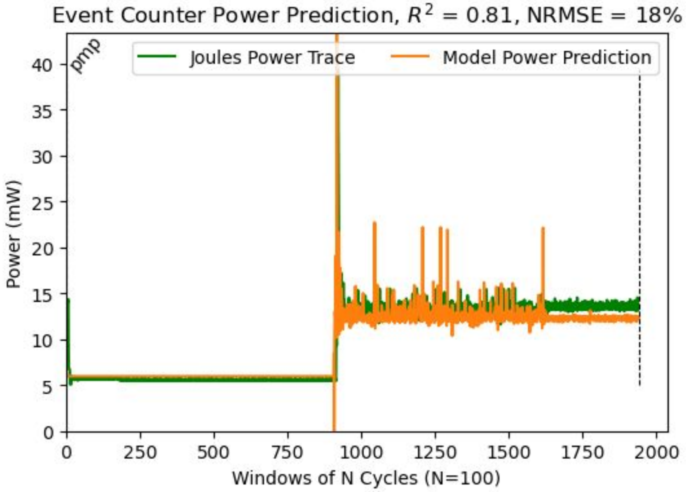
- **Machine Learning for Power**: Nikolic Group
- **Novel Self-Supervised Learning for Circuit Representation Learning:** Wawrzynek Group

ML-assisted Power Modeling

- **Nayiri K. (advised by Prof. Bora Nikolic)**
- Lack of standard benchmarks for both RTL + workloads
    - SOTA reports good errors, but on very different (& usually simple) test data
    - lack of *diversity* in benchmarks
    - need benchmarks that are minimal to avoid wasting compute resources
- Need a power model…
    - that is **auto-generated** (no human input) and **transferable** to…
        - new workloads
        - new RTL? significantly more challenging, not very useful beyond Arch-level
    - with good time & space granularity
- **Power models must have good fidelity across design stage abstractions**
  (Arch → RTL → Syn → PAR)

ML-assisted Power Modeling

- **Goal: input stimuli-dependent dynamic power prediction**
  - Input features: arch counter *deltas* every N cycles
  - Output: predicted power for a window of N cycles
- Golden power
  - .fsdb’s through Joules with Intel16
- Quadratic regression model + ElasticNet(CV)



	R^2		NRMSE
workload	train	test	test .
dhrystone	0.93	0.15	38
median	0.93	0.70	31
mm	0.93	0.42	49
mt-matmul	0.93	-0.64	71
mt-vvadd	0.92	-12.63	41
multiply	0.93	0.44	24
pmp	0.92	0.81	18
qsort	0.92	0.72	26
rsort	0.91	0.21	43
spmv	0.92	0.31	40
towers	0.92	0.21	59
vvadd	0.93	0.72	36

Hierarchical Circuit Representation Learning

- **Josh K. (advised by Prof. John Wawrzynek)**

1) Pretrain a generalizable model on a larger set of representative circuits towards building a single model that can be fine-tuned to perform various downstream tasks

- One large pretrained model ➢ finetune to any EDA task
  - Instead of designing and training small models for individual EDA tasks
  - Better performance even with smaller labeled dataset

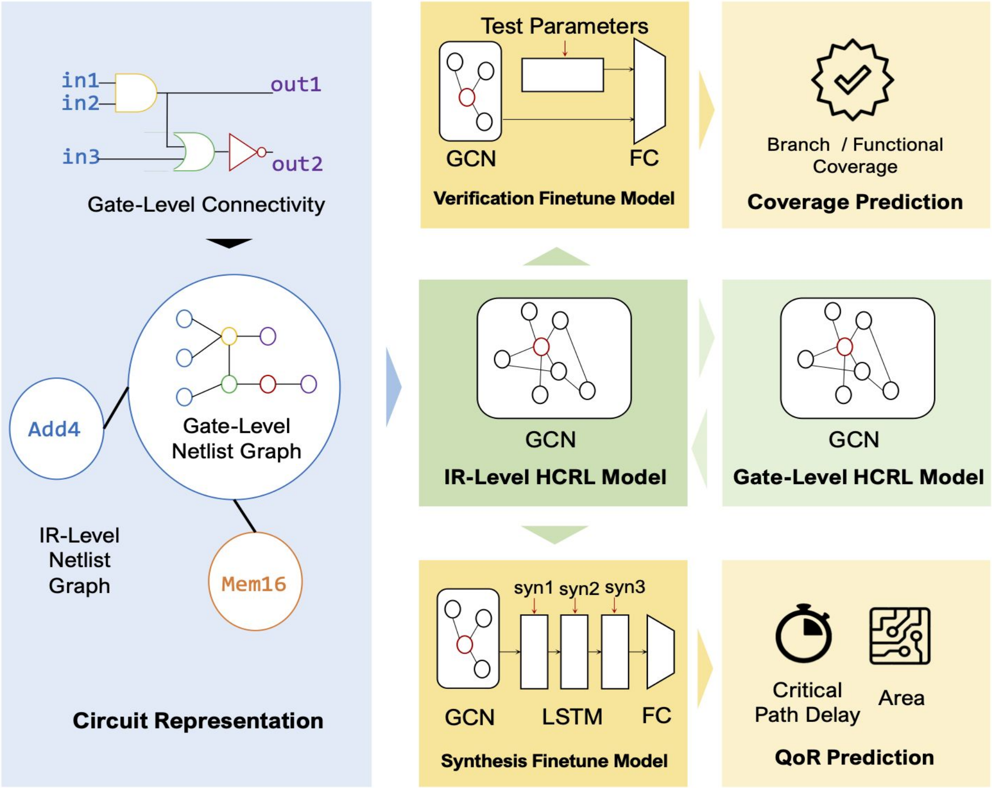
2) Leverage intrinsic graph characteristics that are unique to netlists

- **Hierarchical nature** of circuit netlists:
  - ○ module-level, word-level, gate-level netlists
- **Well-defined semantics for graph similarity**

  ➢ logically equivalent graphs should have similar representations

  Demonstrated by Wang et al. Functionality Matters in Netlist Representation Learning (DAC ‘22)

Hierarchical Circuit Representation Learning



Questions

Berkeley
UNIVERSITY OF CALIFORNIA