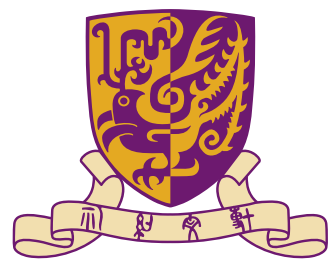
A Learning Bridge from Architectural Synthesis to Physical Design for Exploring Power Efficient High-Performance Adders

Subhendu Roy[1] Yuzhe Ma[2] Jin Miao[1] **Bei Yu**[2]

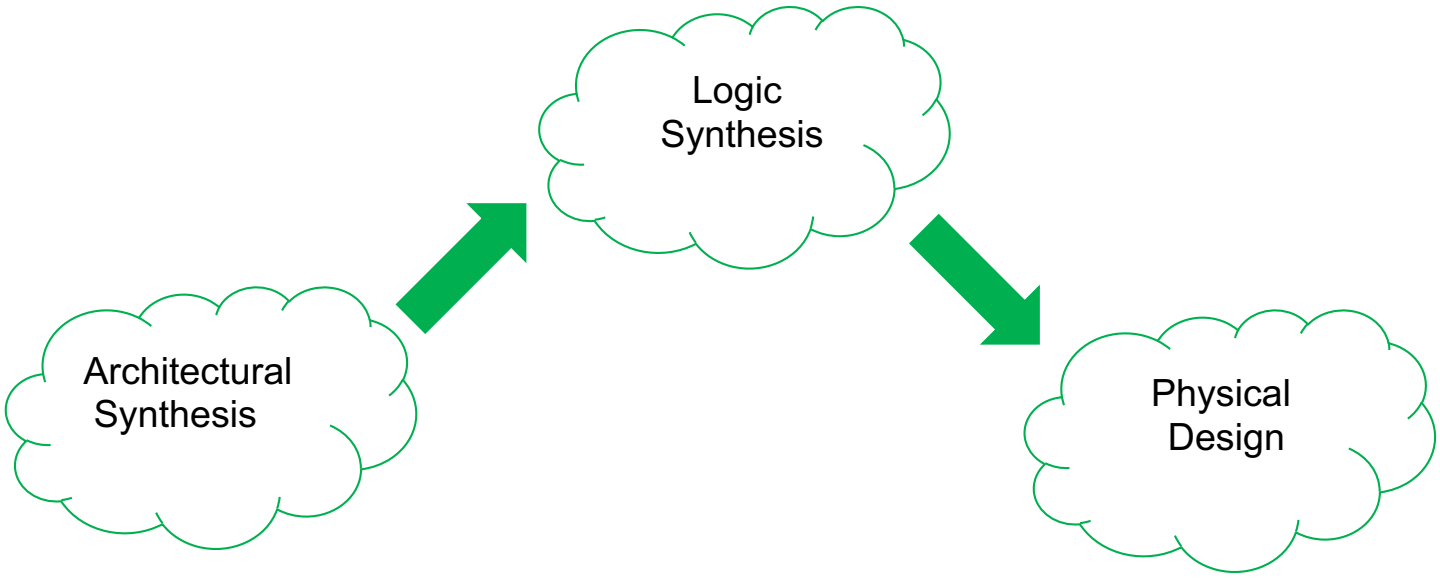
[1]Cadence Design Systems
[2]The Chinese University of Hong Kong

cādence



ISLPED'17

Optimality across EDA stages



No 1-1 mapping between metrics across various EDA stages.

- Optimality at one stage doesn't guarantee the same in another stage
- Data-driven methodology, such as machine learning, becomes imminent

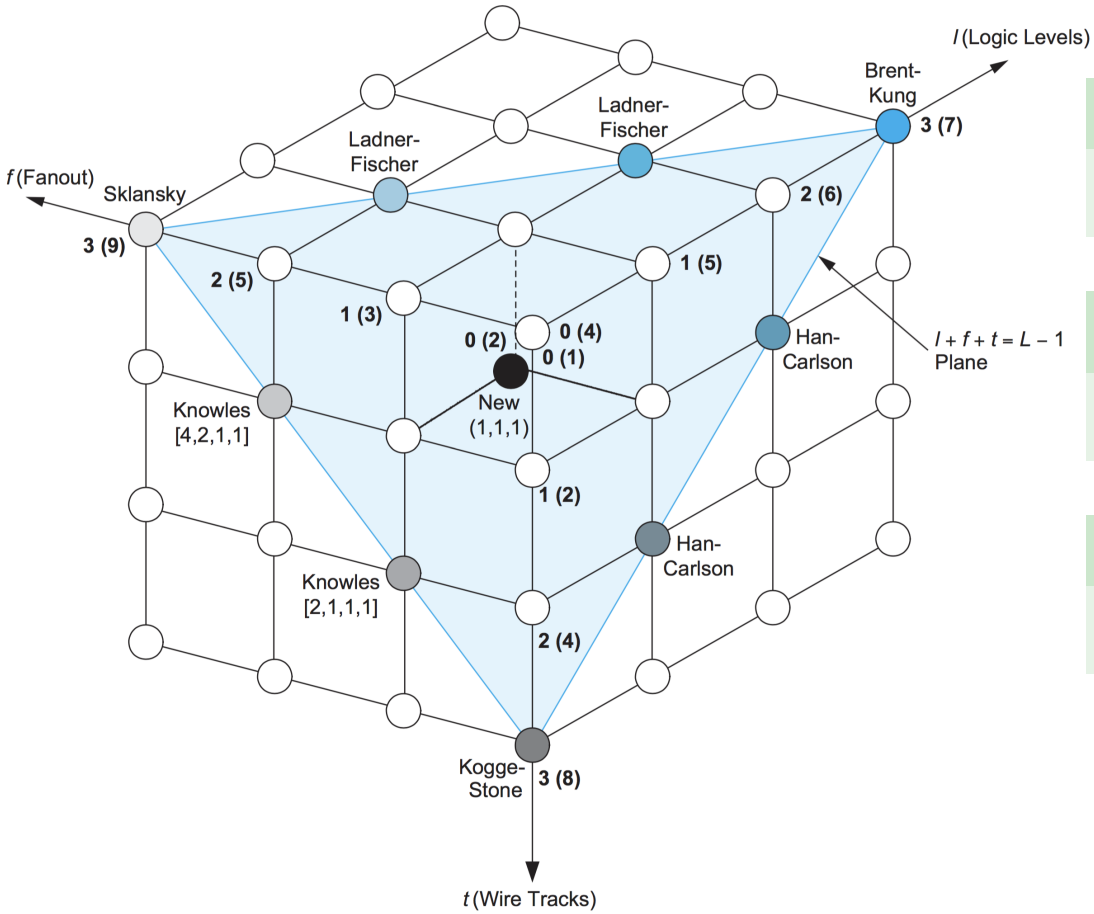
Binary Adder Design

- Primary building blocks in the datapath logic of a microprocessor
- A fundamental problem in VLSI industry for last several decades

What is still unsolved?

Closing the gap across adder design stages

Parallel Prefix Adders



Parallel Prefix Adders

→ Flexible delay-power trade-off

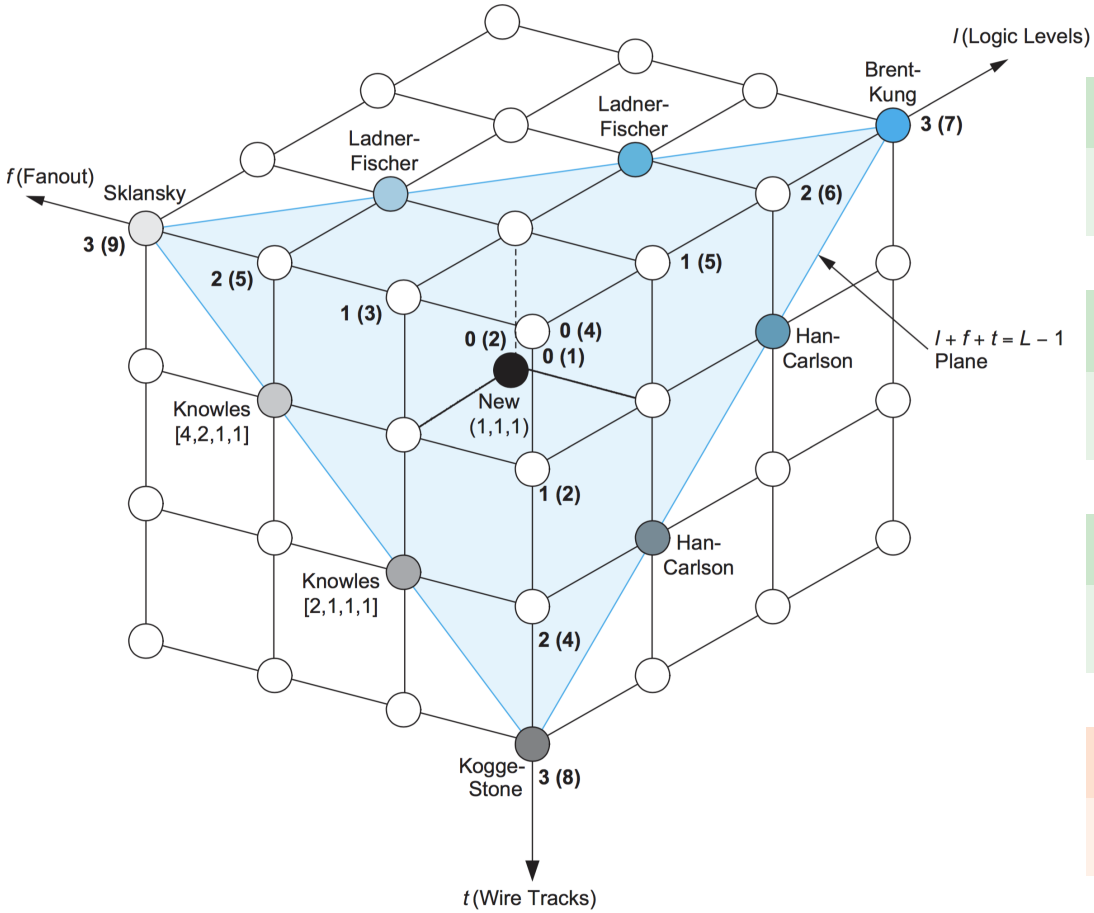
Regular Adders

→ Sub-optimal

Custom Adders

→ High TAT

Parallel Prefix Adders



Parallel Prefix Adders

→ Flexible delay-power trade-off

Regular Adders

→ Sub-optimal

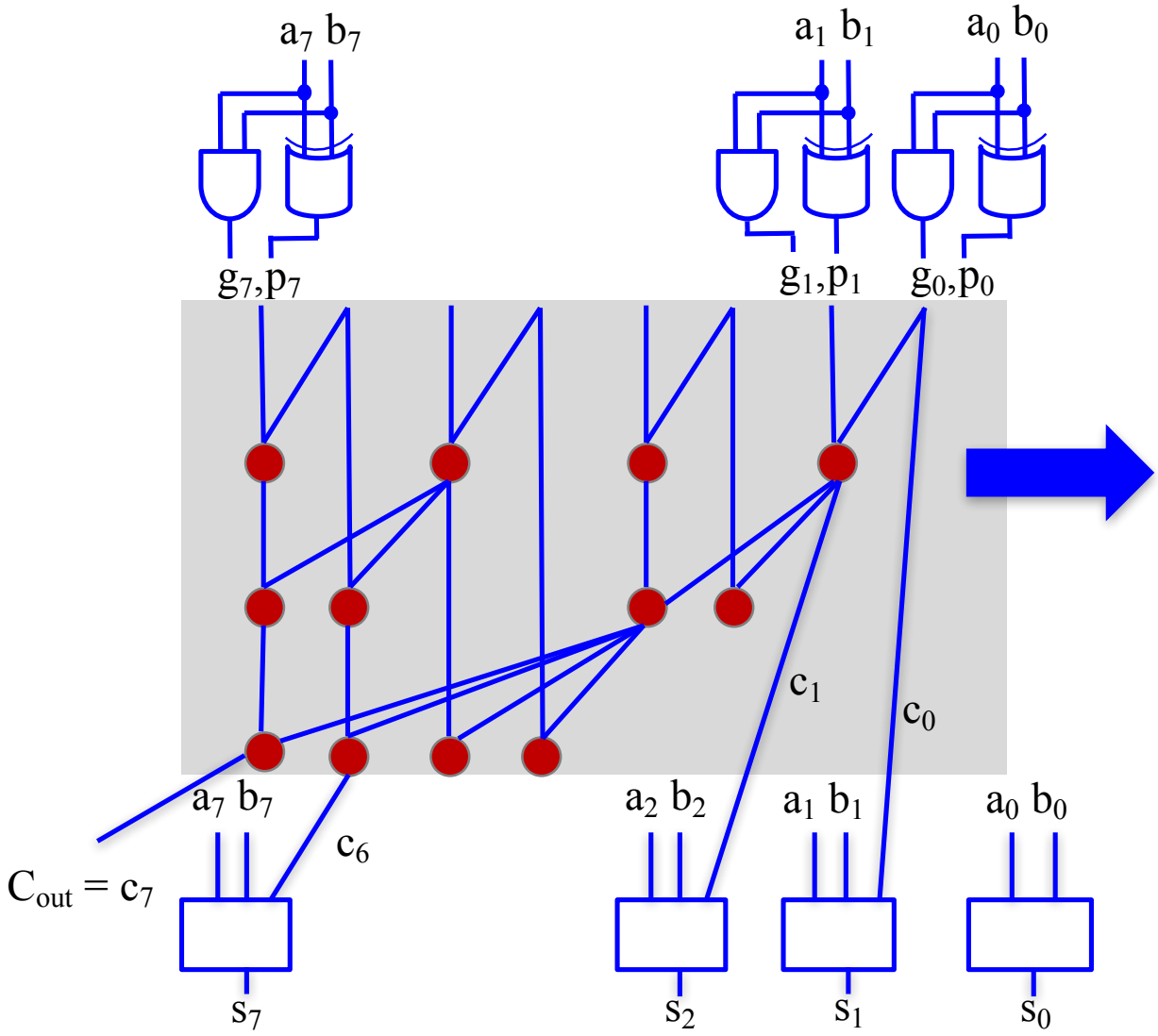
Custom Adders

→ High TAT

This Work:

Automatic Cumtom Adders

Architectural Level: Mapped to Prefix Structures



Pre-processing

Parallel Prefix Structure

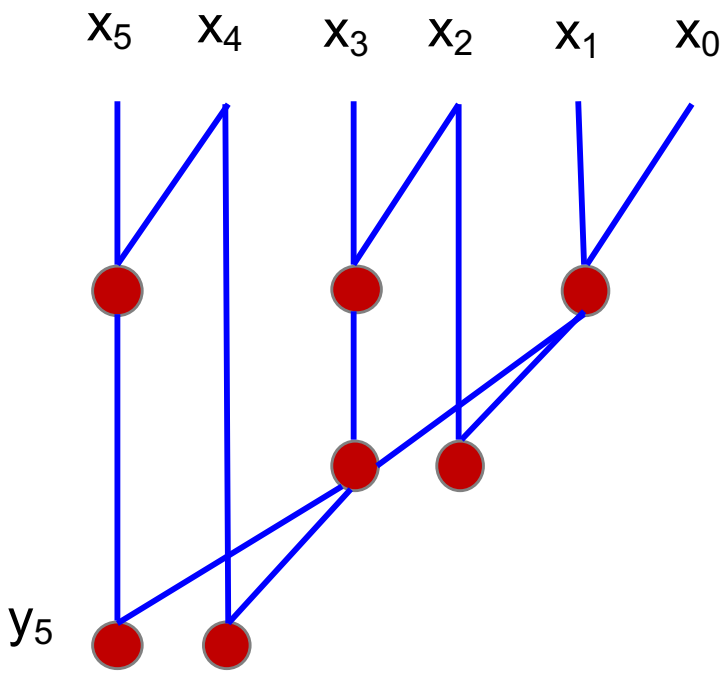
Post-processing

ISLPED'17

Prefix Graph Problem

Carry-computation can be mapped to prefix graph problem

$$\mathbf{y}_i = \mathbf{x}_i - 1 \; o \; \mathbf{x}_{i-1} \; o \; \mathbf{x}_{i-2} \; o \ldots \mathbf{x}_1 \; o \; \mathbf{x}_0$$



Size (s) = No. of prefix nodes = 7
Level (L) = maximum logic level = 3
Max-Fanout (mfo) = 2

Classifying Prefix Graph Synthesis

Can be classified based on the solution#

Category 1: Limited number of solutions

- Example: [Matsunaga+,GLSVLSI'07], [Liu+,ICCAD'03], [Zhu+,ASPDAC'05], [Roy+,ASPDAC'15]

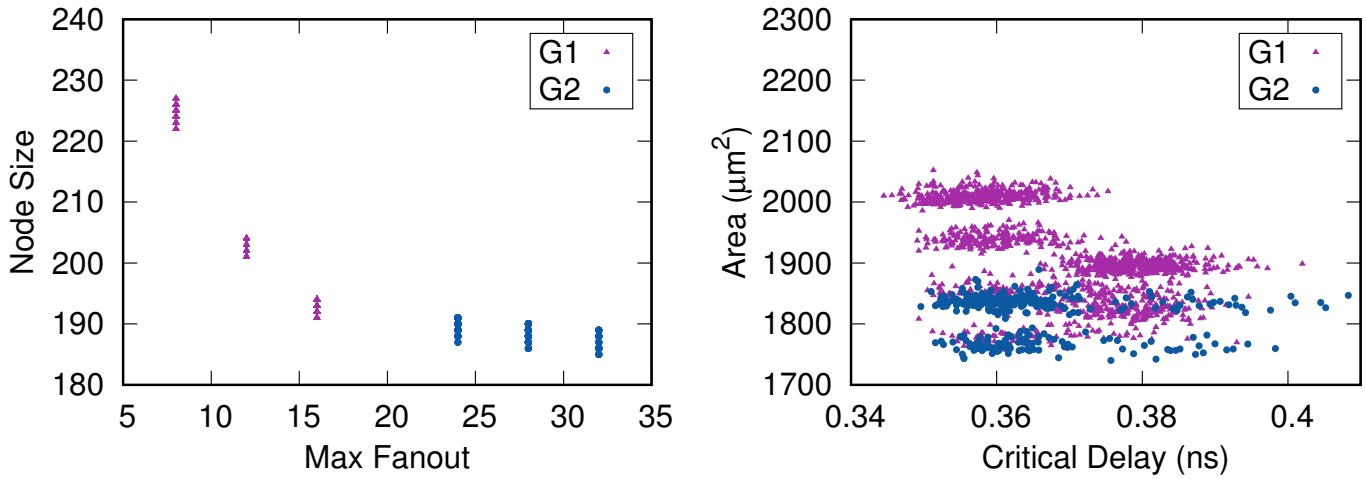
- Not suitable for exploring data-driven methodologies
- No analytical model to physical design stage

Category 2: Innumerable solutions

- Example: [Roy+,TCAD'14]

- Not scalable for bounded fan-out
- Computationally expensive to run all solutions through full physical design flow

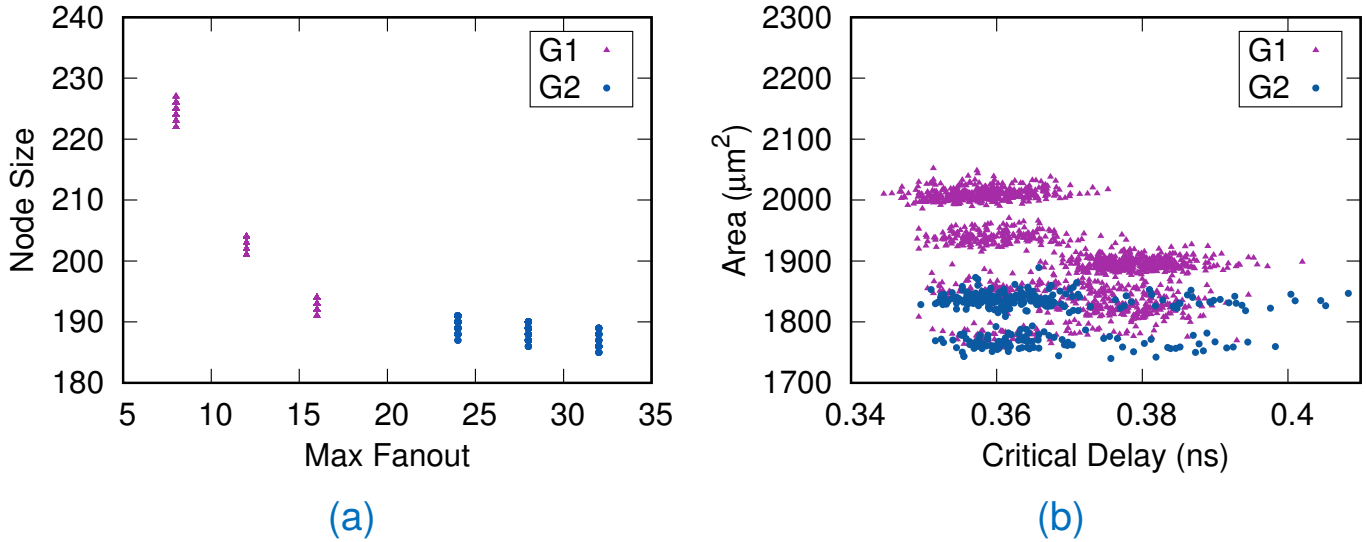
Gap between Prefix Structure and Physical Design



(a) Architectural solution space; (b) Physical design space.

- G1 (less fan-out and high size); G2 (high fan-out and low size)
- When mapped to physical solution space
  - Correlation between size and area
  - Not completely reliable, G1 and G2 get mixed up in physical solution space

Gap between Prefix Structure and Physical Design



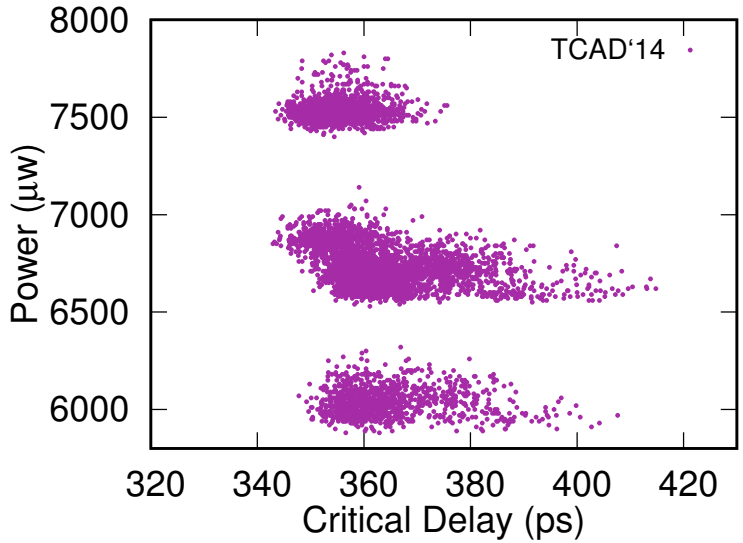
(a) Architectural solution space; (b) Physical design space.

- G1 (less fan-out and high size); G2 (high fan-out and low size)
- When mapped to physical solution space
  - Correlation between size and area
  - Not completely reliable, G1 and G2 get mixed up in physical solution space

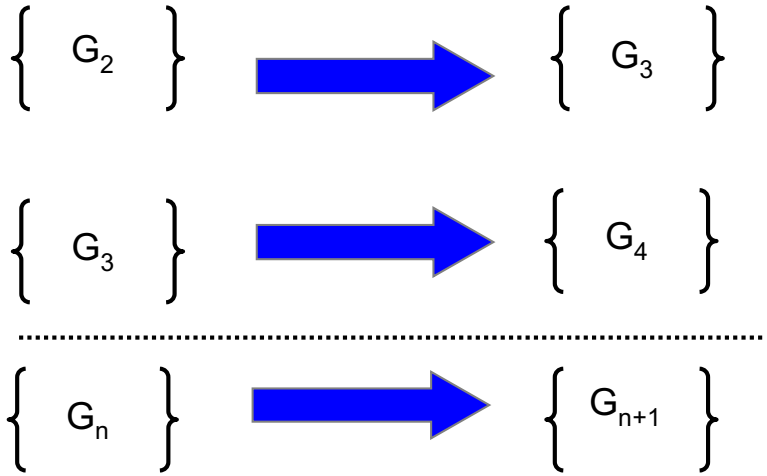
What We Want to Search For:

All Pareto Frontier points with low area, low power, and low critical delay.

Task 1: Prefix Adder Solution Exploration



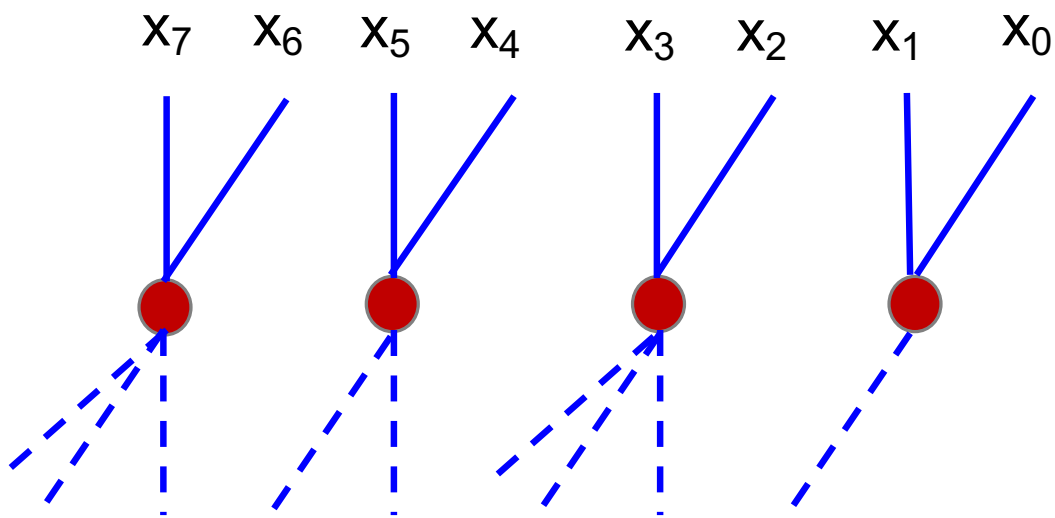
[Roy+,TCAD'14]– Summary



- G_n = set of prefix graphs of bit-width n
- Prefix graphs of higher order generated in bottom-up fashion
- Several pruning strategies during $G_n \rightarrow G_{n+1}$ for scaling
  - For bounded fan-out, these strategies compromises in size-optimality

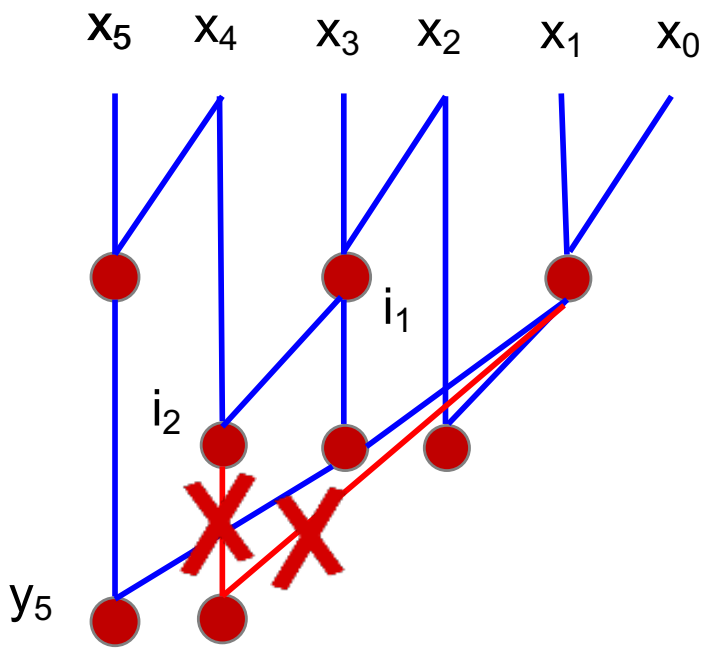
Enhancement 1: Imposing Semi-regularity

- The concept is derived from regular adders such as Brent-Kung, Sklansky.
- x_i and x_{i+1} combined to form prefix nodes, where i is even.
- This regularity for only $L = 1$
- For $L > 1$, regularity compromises size optimality (Forbidden).
- Observation: this semi-regularity doesn't degrade size-optimality.



Enhancement 2: Level restriction in Non-trivial Fan-in

- Trivial fan-in having same MSB
- x_4 and i_1 are trivial and non-trivial fan-in of i_2
- Level (non-trivial fan-in) $\geq$ level (trivial fan-in)
- Reduces search space without degrading size-optimality



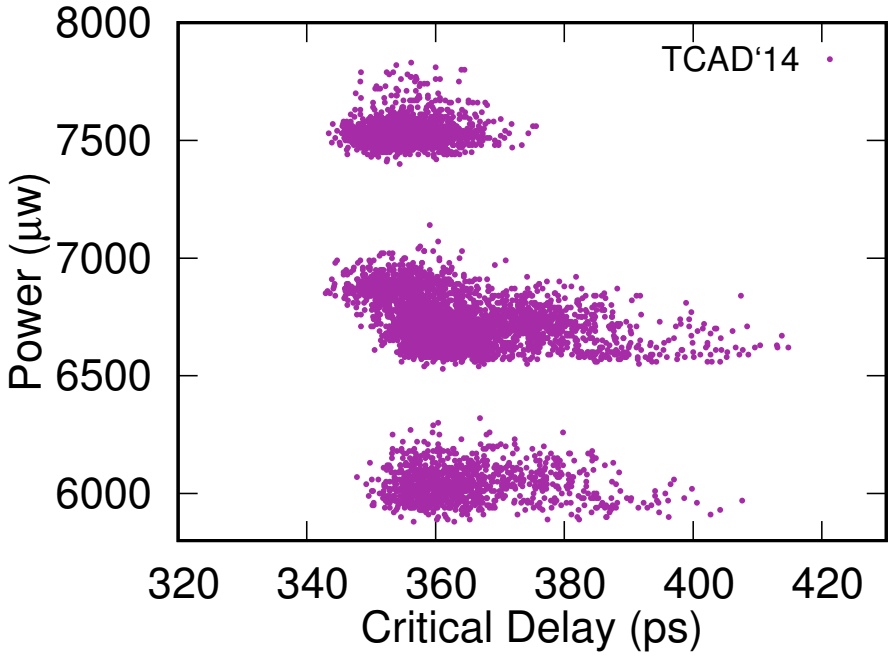
ISLPED'17

Comparison at Prefix Graph Stage

mfo	Our Approach		[Roy+,TCAD'14]	
	size	Run-time (s)	size	Run-time (s)
4	244	302	252	241
6	233	264	238	212
8	222	423	-	-
12	201	193	-	-
16	191	73	192	149
32	185	0.04	185	0.04

- Table is for 64 bit adders
- [Roy+,TCAD'14] cannot get solutions for all fanouts.
- Our solutions are always more size-optimal.
- Runtimes are comparable, adder synthesis is one-time.

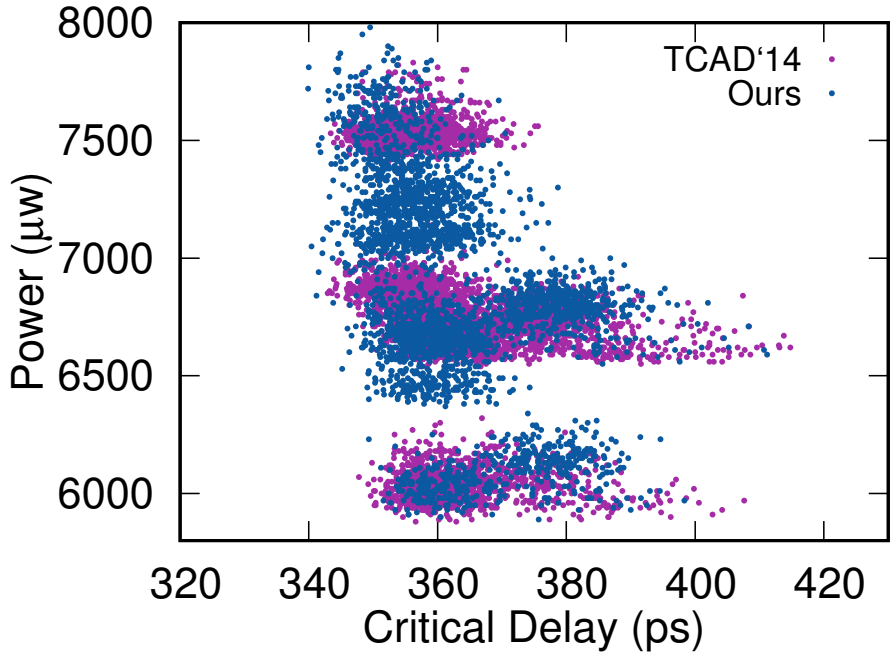
Physical Solution Space Comparison



Our solutions cover wider space in physical domain

- 7000 random samples from [Roy+,TCAD'14] vs. 3000 samples from us
- Reason: TCAD'14 misses solutions for bounded fanout in a few cases

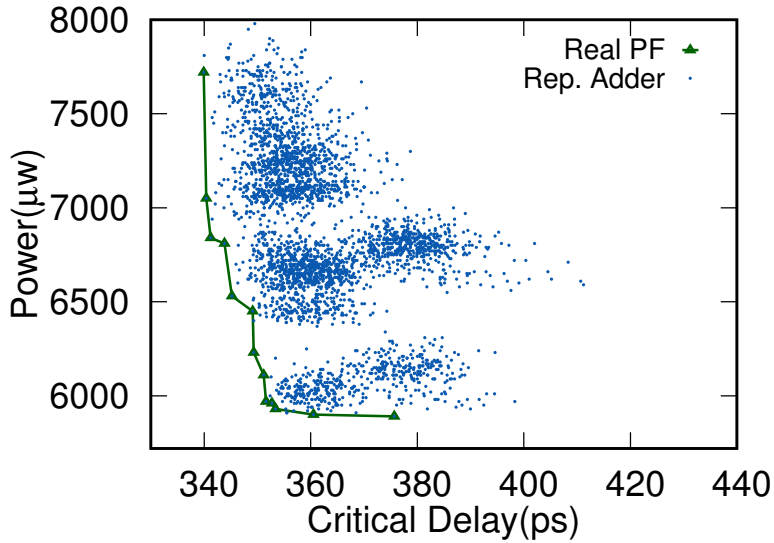
Physical Solution Space Comparison



Our solutions cover wider space in physical domain

- 7000 random samples from [Roy+,TCAD'14] vs. 3000 samples from us
- Reason: TCAD'14 misses solutions for bounded fanout in a few cases

Task 2: Pareto Frontier Driven Learning



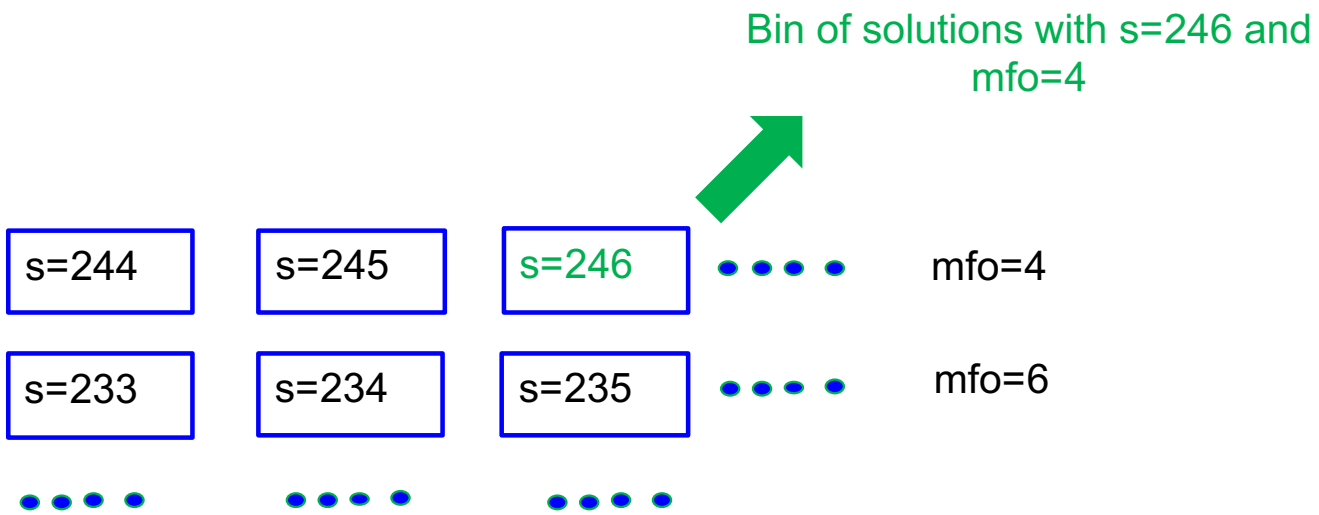
Quasi-Random Data Sampling

- Hundreds of thousands of solutions
- How to choose training data?
  - Cannot run too many architectures as physical design flow costly.
  - Too few will degrade model accuracy.

Quasi-Random Sampling

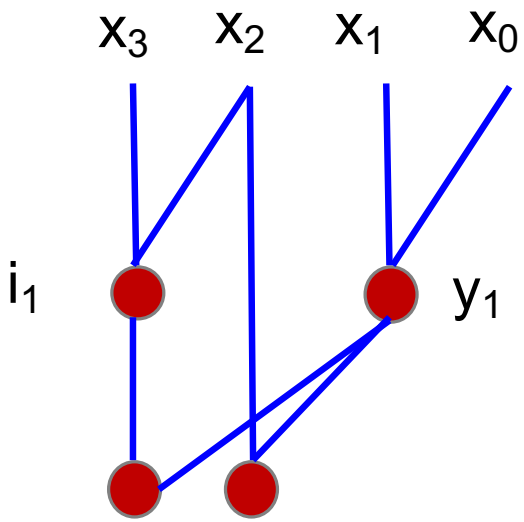
Create architectural bins based on mfo and s.

- Capture all architectural bins
- Select solutions from each bin randomly



Feature Selection and Learning Model

- Architectural attributes: s, mfo, sum-path-fanout ($spfo$)
- Tool settings: Target delay
- Best model fitting by support-vector-regression (SVR) with RBF kernel
- Including $spfo$ improves MSE score for delay from 0.232 to 0.164
- Note: linear models not sufficient for modeling delay



$$spfo(y1) = spfo(x0) + spfo(x1) + fo(x0) + fo(x1) = 0 + 0 + 1 + 1 = 2$$

$$spfo(i1) = spfo(x3) + spfo(x2) + fo(x3) + fo(x2) = 0 + 0 + 1 + 2 = 3$$

$$spfo(y3) = spfo(i1) + spfo(y1) + fo(i1) + fo(y1) = 3 + 2 + 1 + 2 = 8$$

Pareto Frontier Driven Learning

- Conventional learning focusses on prediction accuracy
  - Model accuracy improvement doesn't guarantee Pareto-frontier improvement
  - Need for learning integrated Pareto-frontier exploration
- Scalarization or α-sweep
  - Learning output is a linear sum of delay and power ($\alpha \times$ Power + Delay)
  - Model-fitting done with different values of alpha
  - Sweeping alpha from 0 to a large positive number

Experimental Setup

Synthesis and placement/routing of adders

- Tools: Design Compiler/ IC Compiler
- Library: Non-linear-delay-model (NLDM) in 32nm SAED cell-library
- Tool settings: Target delay = 0.1ns, 0.2ns, 0.3 ns

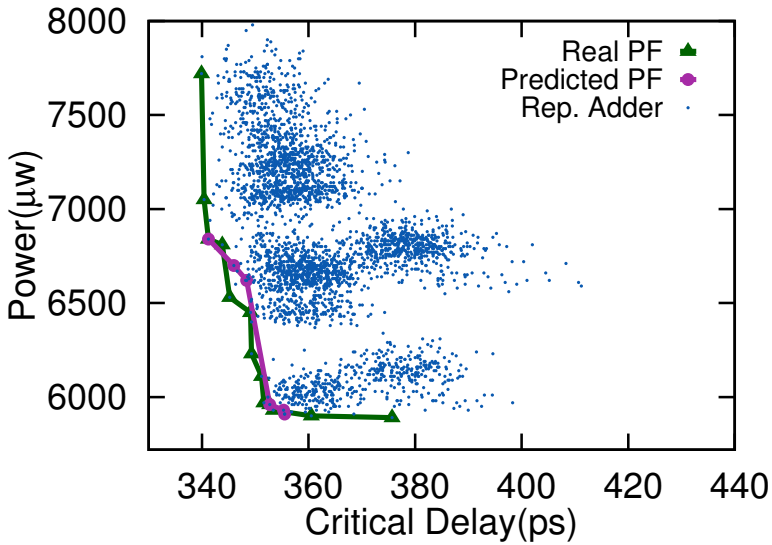
Programming Language

- C++ for prefix adder synthesis
- Python based machine learning package scikit-learn

Machine Configurations

- 72GB RAM UNIX machine
- 2.8GHz CPU

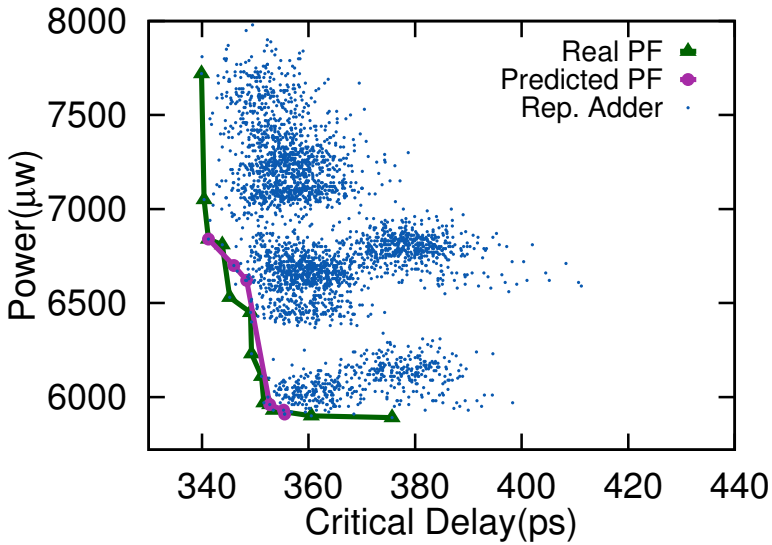
Pareto-frontier Comparison

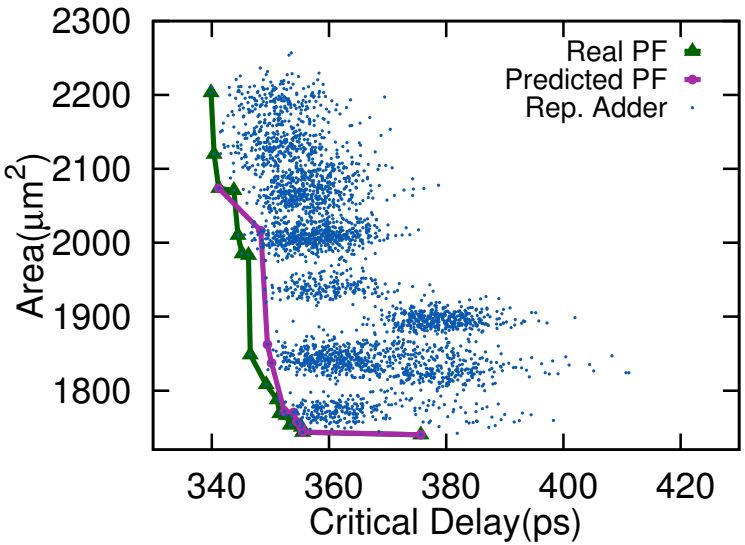


Predicted pareto-frontier almost matches actual pareto-frontier

- Training set is randomly selected from 300 samples.
- Rep. adders are quasi-random sampled from other 3000 samples
- Predicted frontier is from best 150 solutions (predicted)

Pareto-frontier Comparison





Predicted pareto-frontier almost matches actual pareto-frontier

- Training set is randomly selected from 300 samples.
- Rep. adders are quasi-random sampled from other 3000 samples
- Predicted frontier is from best 150 solutions (predicted)

Comparison with Other Adders

Pareto-points derived from our approach beats other solutions in all metrics (delay, area, power)

Method	Delay (ps)	Area (μm^2)	Power (mW)
Kogge-Stone	347.9	2563.7	8.78
Ours (P_1)	340.0	2203.3	7.72
Sklansky	356.1	1792.5	6.1
Ours (P_2)	353.0	1753.0	5.9
[Roy+,ASPDAC'15]	348.7	1971.4	6.98
Ours (P_3)	346.0	1848.6	6.67

Conclusion

Machine learning guided design space exploration

- For power-efficient high-performance adders
- Bridge the gap between architectural and physical solution space
- Provide near-optimal power vs. delay trade-off

Our methodology excels

- State-of-the-art adder synthesis algorithms in power/delay/area metrics
- Readily adoptable for any cell-library

Thank You

Subhendu Roy (subhroy@cadence.com)

Yuzhe Ma (yzma@cse.cuhk.edu.hk)

Jin Miao (jmiao@cadence.com)

Bei Yu (byu@cse.cuhk.edu.hk)

cadence