Efficient Synthesis Methods for High-Quality Approximate Computing Circuits

Chang Meng, *chang.meng@epfl.ch* Advisor: Prof. Weikang Qian
Obtained Ph.D. degree in June 2023 from Shanghai Jiao Tong University, China
Current position: postdoctoral researcher at EPFL, Switzerland

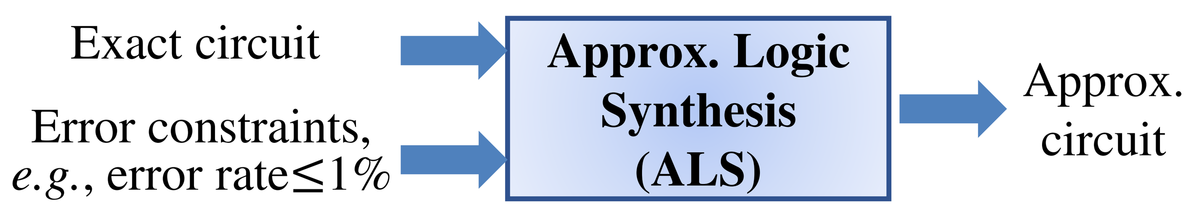
Approximate Logic Synthesis — Generating Approx. Circuits that Trade Accuracy for Quality

Approximate computing

- Improve circuit quality by sacrificing accuracy
- Target at error-tolerant applications:

Image processing *Data mining* *Neural network*

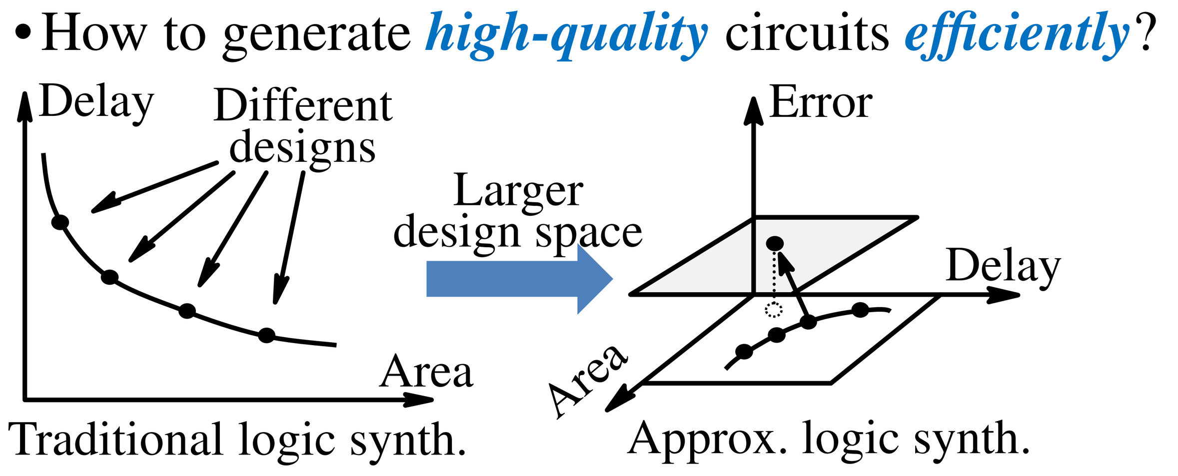
Approximate logic synthesis (ALS)



- Objective: improve **quality** of approximate circuits, *i.e.*, *small area, delay, and power*
- Satisfying the error constraints

Large design space brings challenges

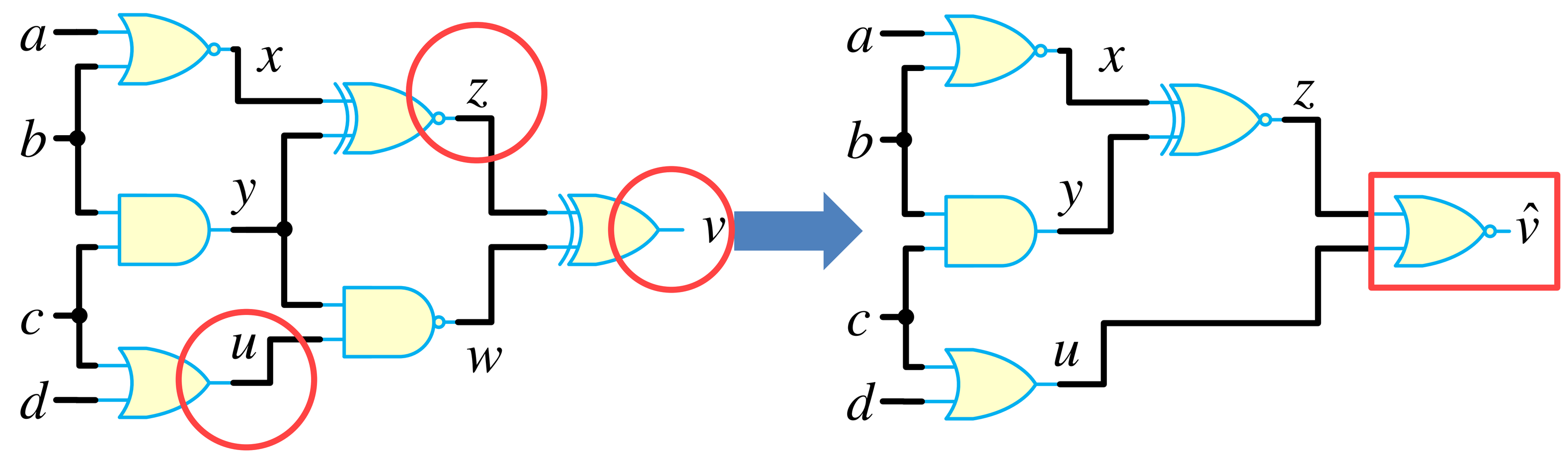
- How to generate ***high-quality*** circuits ***efficiently***?



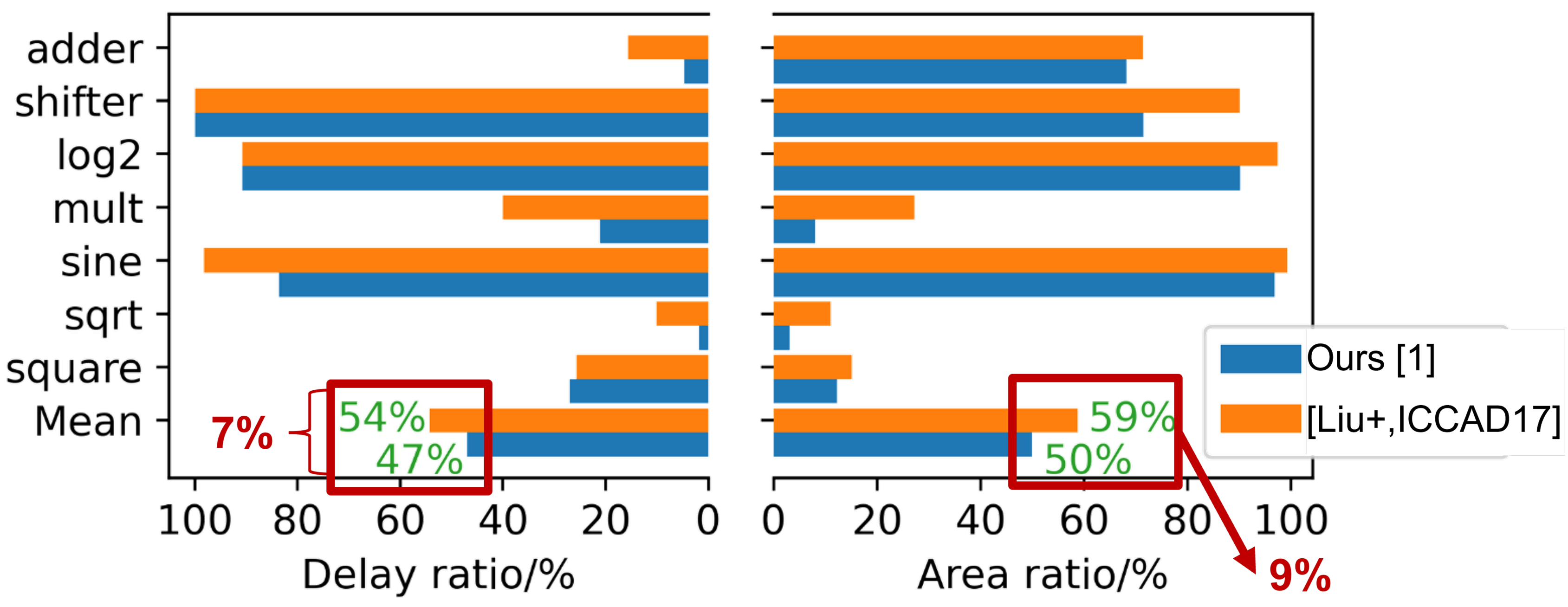
Improve Circuit Quality — Area-Oriented & Delay-Oriented Approximate Logic Synthesis

Area-Oriented ALS by Approximate Resubstitution [1]

- Approximately re-express node v with a set of other nodes $\{u, z\}$
- Error rate = 3/16 → ***Remove w & simplify v from XOR to NOR gate***

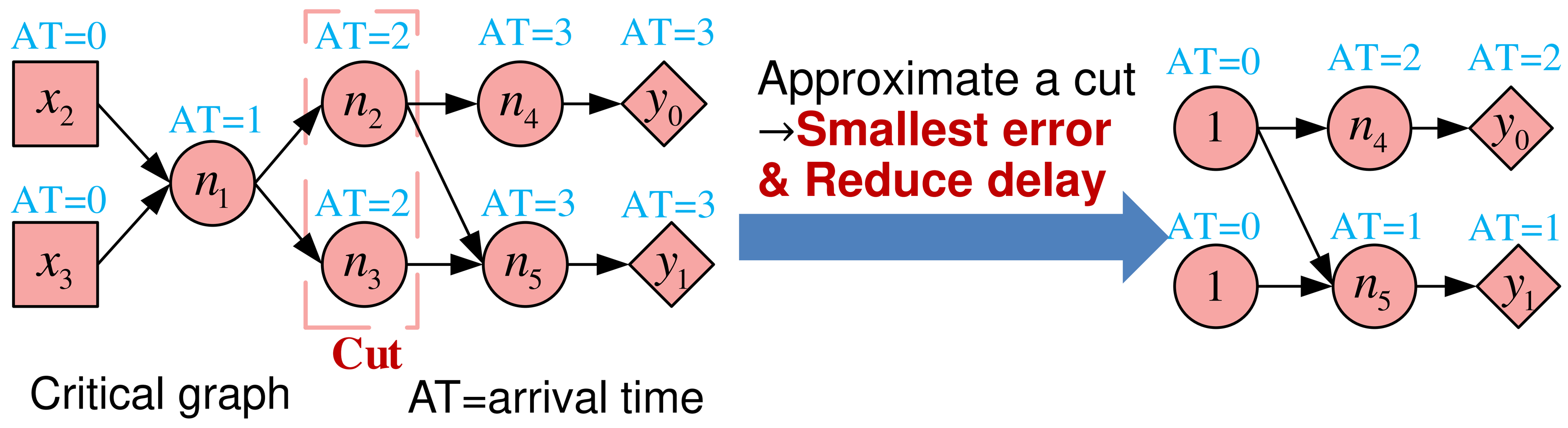


- Iteratively apply AppResubs until reaching error threshold
- Under normalized mean error distance $\leq 0.2\%$:



Delay-Oriented ALS Targeting at Critical Paths [2]

- Collect all critical paths to form a **critical graph**:



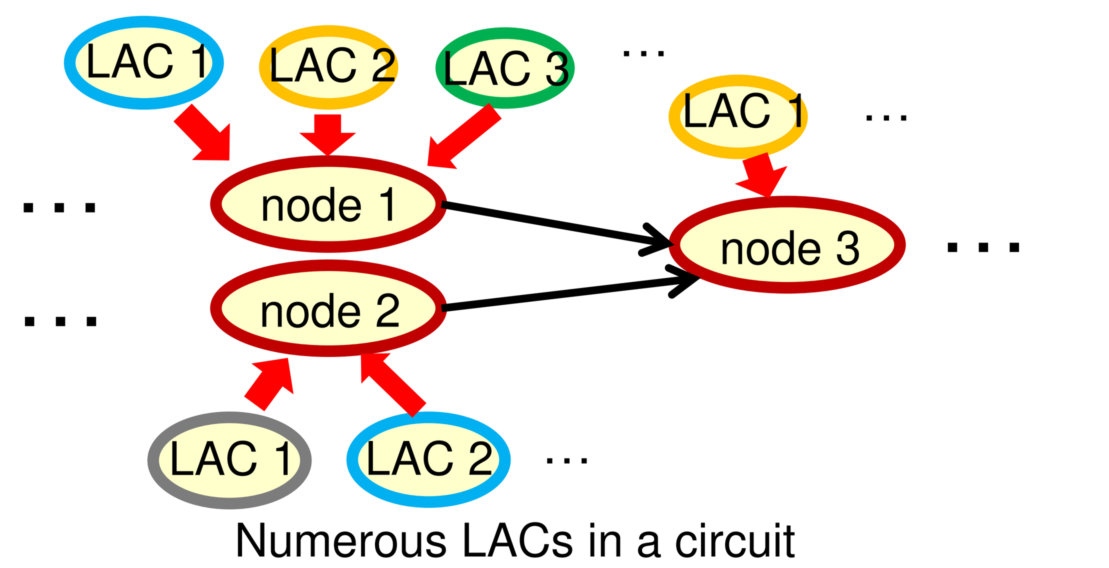
- Iteratively selecting ***min-error cut*** & ***approximate all nodes*** in the cut:

Circuit	NMED bound	Delay ratio Ours	Delay ratio [Ma+, 2021]	Area ratio Ours	Area ratio [Ma+, 2021]
absdiff	0.59%	**65.9%**	103.9%	**73.4%**	83.5%
	2.94%	**42.3%**	98.3%	**40.2%**	74.7%
adder32	0.59%	**30.5%**	43.6%	40.2%	31.9%
	2.94%	**18.9%**	24.3%	**29.3%**	33.1%
buttfly	0.59%	**36.0%**	45.7%	**43.7%**	61.7%
	2.94%	**25.3%**	37.5%	**31.5%**	41.2%
mac	0.59%	**92.1%**	97.6%	96.3%	95.2%
	2.94%	**53.4%**	104.8%	**48.8%**	72.4%
mult8	0.59%	**57.8%**	60.7%	78.2%	34.5%
	2.94%	**11.7%**	34.0%	**14.0%**	16.2%
mult16	0.59%	**59.2%**	70.6%	37.8%	15.8%
	2.94%	**29.6%**	52.1%	**7.6%**	7.8%
Average	-20.8%!	**43.6%**	64.4%	**45.1%**	47.3%

Enhance ALS Efficiency — Efficient Estimation of Average & Maximum Error Metrics

Runtime Bottleneck: Error Estimation

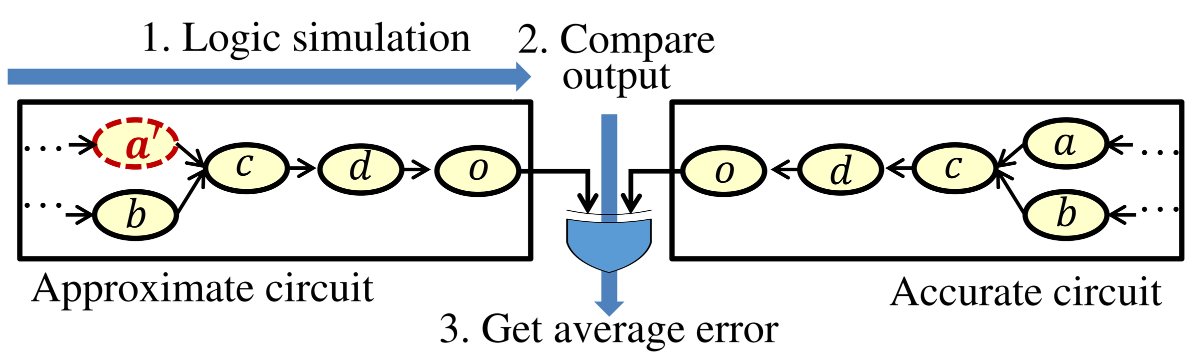
- **Local approximate change (LAC)**: local structure modification to simplify the circuit, *e.g.*, constant replacement, AppResub [1], etc.



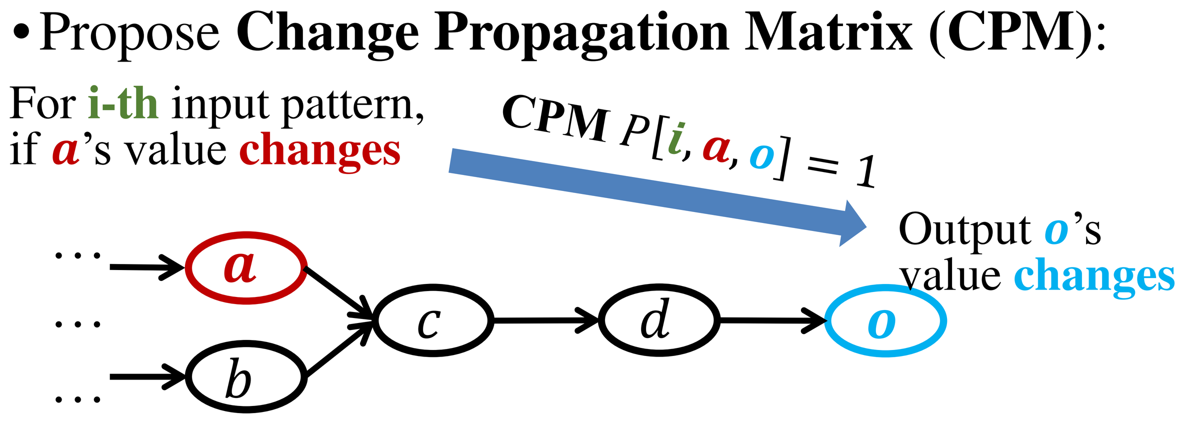
- Meet error constraints→***evaluate all LACs' errors***
- Two kind of errors:
  - **Average error**: average deviation between approximate & exact outputs, *e.g.*, error rate, mean error distance, etc.
  - **Maximum error**: maximum deviation between outputs of approximate & exact circuits, *e.g.*, maximum error distance, etc.

Fast Average Error Estimation [3]

- Previously, ***simulate circuit for each LAC!***



- Propose **Change Propagation Matrix (CPM)**:

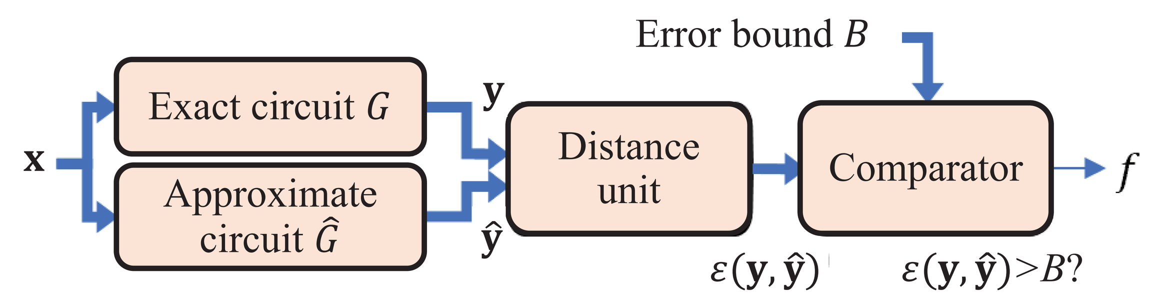


Linear-complexity **recursive update**:
$P[i,a,o] \Leftarrow P[i,c,o] \Leftarrow P[i,d,o] \Leftarrow P[i,o,o] = 1$

- CPM⇒values of circuit outputs⇒average error
  - ***Avoid*** simulating the circuit for each LAC!
  - Accelerate by **110 ×**!

Fast Maximum Error Checking [4]

- Previously, ***solve SAT problem for each LAC!***



f* is UNSAT ⇔ max. error ≤ *B

- Different LACs have ***similar*** SAT instances.
- Merge similar SAT instances into one & Perform ***single SAT sweeping***!

Circuit	Mean area saving MUSCAT	Mean area saving MECALS	Mean delay saving MUSCAT	Mean delay saving MECALS	Runtime/s MUSCAT	Runtime/s MECALS
absdiff	47.0%	**67.6%**	26.6%	**42.5%**	67	**3**
add8	55.9%	**58.9%**	45.2%	**53.3%**	15	**1**
add32	40.3%	**47.7%**	45.8%	28.4%	29	611
binsqrd	9.8%	**27.0%**	5.5%	**8.2%**	31546	**791**
buttfly	15.1%	**24.0%**	4.4%	**17.7%**	8	51
mac	10.0%	**32.2%**	5.7%	**13.9%**	3	8
mult8	61.5%	**75.8%**	38.7%	**53.3%**	7380	**1548**
	-13.4% area		-6.4% delay		13x faster	
Mean of the above	34.2%	**47.6%**	24.6%	**31.0%**	5578	**431**

Thesis Contributions

Improve Approx. Circuits' Quality

Area-oriented ALS: reduce **9% area**
Delay-oriented ALS: reduce **20.8% delay**

Enhance ALS Efficiency

Fast average error estimation: accelerate by **110 times**
Fast maximum error checking: accelerate by **13 times**

References

[1] C. Meng, W. Qian, and A. Mishchenko, "ALSRAC: Approximate logic synthesis by resubstitution with approximate care set," in *DAC*, 2020, pp. 1–6.
[2] C. Meng, Z. Zhou, Y. Yao, S. Huang, Y. Chen, and W. Qian, "HEDALS: Highly efficient delay-driven approximate logic synthesis," *IEEE TCAD*, 2023.
[3] S. Su, C. Meng, F. Yang, *et al.*, "VECBEE: A versatile efficiency–accuracy configurable batch error estimation method for greedy approximate logic synthesis," *IEEE TCAD*, 2022.
[4] C. Meng, J. Sun, Y. Mai, and W. Qian, "MECALS: A maximum error checking technique for approximate logic synthesis," in *DATE*, 2023, pp. 1–6.

My website&codes