Rank-based Multi-objective Approximate Logic Synthesis via Monte Carlo Tree Search

Yuyang Ye[1], Xiangfei Hu[2], Yuchen Liu[2], Peng Xu[1], Yu Gong[3], Tinghuan Chen[4], Hao Yan[2], Bei Yu[1], Longxing Shi[2]

[1]CUHK [2]Southeast University [3]Nanjing University of Aeronautics and Astronautics [4]CUHK-Shenzhen

62 THE CHIPS TO SYSTEMS CONFERENCE

SPONSORED BY CEDA sigda

Outline

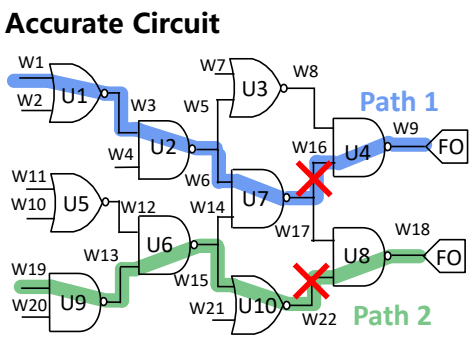
Introduction

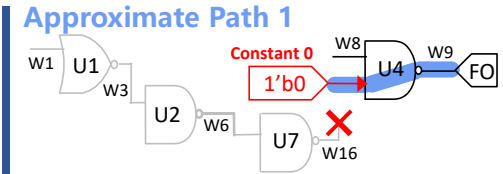
62 THE CHIPS TO SYSTEMS CONFERENCE

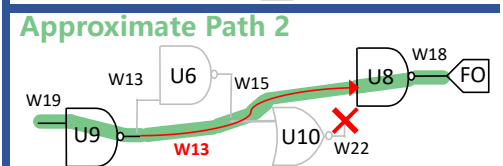
SPONSORED BY CEDA IEEE Council on Electronic Design Automation sigda ACM

Approximate Logic Synthesis

Approximate logic synthesis can bring **timing** and **area** improvements under error constraints through two state-of-the-art local approximate changes (LACs), including **wire-by-constant** and **wire-by-wire** replacements.







Error Constraints

Error distance. The difference between the approximate and accurate circuit output values under one input vector.
Error rate. The percentage of input vectors that the approximate circuit output differs from the exact circuit.

$$\mathrm{NMED} = \sum_{i=1}^{2^I} \frac{P_i \times \left| V_i^{\mathrm{app}} - V_i^{\mathrm{acc}} \right|}{2^O - 1}. \tag{1}$$

$$\mathrm{ER} = \sum_{i=1}^{2^I} \left[P_i \times (O_i^{\mathrm{app}} \neq O_i^{\mathrm{acc}}) \right]. \tag{2}$$

Existing Methods

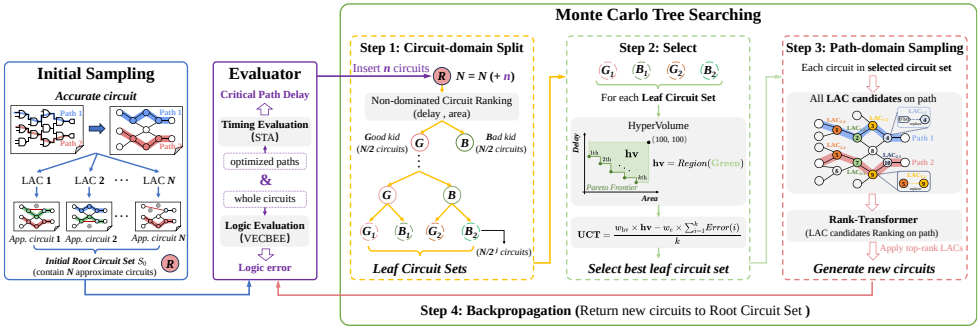
- Timing-driven Methods perform LACs to simplify gates on critical paths.
- Area-driven Methods iteratively select LACs with great area reduction potential. These reductions can be converted into drive strength enhancement of gates.

Algorithms

SPONSORED BY CEDA sigda

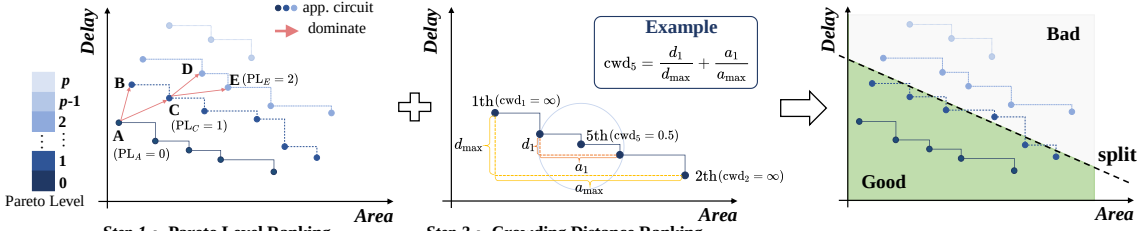
Overall Flow

Our framework uses MCTS to deeply explore the optimization potential of delay and area. Each node in the search tree represents a circuit set, containing multiple approximate circuits. The MCTS receives initial circuits and iteratively performs four steps: *Split*, *Selection*, *Sampling* and *Backpropagation*.



Circuit-domain Rank-based Splitting

The split step is designed to achieve the partitioning of the current root circuit set $S_t : \{c_1^t, c_2^t, \ldots, c_n^t\}$ and reconstruct a performance-driven search tree.



Selection

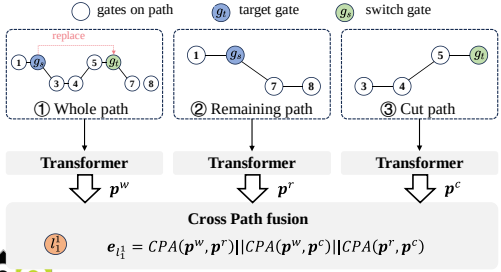
Selection filters the best circuit set based on the upper confidence bound for trees (UCT) from the leaf circuit sets of the search tree generated after splitting. Since ALS requires sufficient optimization of the objectives with minimal error, UCT is computed based on the HyperVolume **hv** and the average error of k Pareto front circuits within the circuit set.

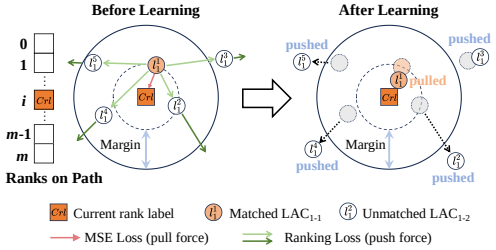
$$\mathbf{hv} = \sum_{i=1}^{k} \left[(1 - Ratio_d^{(i)}) \times (Ratio_a^{(i-1)} - Ratio_a^{(i)}) \right]. \tag{3}$$

$$\mathbf{UCT} = \frac{w_{hv} \times \mathbf{hv} - w_e \times \sum_{i=1}^{k} Error(i)}{k}, \tag{4}$$

Path-domain Rank-based Sampling

The sampling is performed on all circuits inside the selected leaf circuit set. In our MCTS framework, we maintain and apply a set of timing-area-reducing LAC $\mathcal{L}_M : \{l_M^1, l_M^2, \ldots, l_M^n\}$ for each circuit to generate new approximate circuits.





Experimental Results

Expermental settings

- For each generated approximate circuit, we use Synopsys PrimeTime to report timing results. Meanwhile, the circuit area is obtained using ABC. In terms of logic simulation, we randomly generate 100,000 input vectors for VECBEE.
- CPU Device: a 72-core 2.6GHz Linux machine with 1024 GB memory.
- GPU Device: 4 NVIDIA Tesla V100 GPUs.
- Benchmarks: ISCAS'85 and EPFL circuits.

Table: Statistics of the benchmarks used in our experiment. The units of delay and area are respectively ps and μm^2.

Random / Control				Arithmetic			
Circuit	#Gate	Delay	Area	Circuit	#Gate	Delay	Area
c880	322	185.34	177.67	c6288	1641	847.79	687.08
c1908	366	235.14	223.34	adder	1639	1394.7	495.78
c2670	922	218.40	288.71	barshift	2933	262.52	1806.6
c3540	667	293.09	459.42	max	2940	2799.8	954.03
c5315	2595	122.25	1129.6	mult	26429	4117.5	31635.6
c7552	1576	282.13	939.33	sine	11560	3234.4	7173.9
cavlc	573	186.35	450.31	sqrt	13542	67929.3	6262.1
priority	2336	1126.8	1423.3	square	14696	8211.1	7752.8

Optimization Results under ER constraint

Table: Comparison of multi-objective optimization performance between our framework and other works under the 3% *ER* constraints.

Circuit	VECBEE-S[1]		HEDALS[2]		TCAD24		DCGWO		Ours	
	delay ratio	area ratio	delay ratio	area ratio	delay ratio	area ratio	delay ratio	area ratio	delay ratio	area ratio
c880	92.16%	86.75%	93.02%	89.22%	83.21%	84.98%	85.21%	82.76%	**77.09%**	**67.54%**
c1908	84.21%	63.37%	**46.12%**	59.36%	48.15%	62.34%	43.97%	57.02%	49.89%	**45.72%**
c2670	79.14%	69.78%	79.64%	94.17%	75.39%	61.28%	76.92%	60.33%	**74.11%**	**56.96%**
c3540	97.93%	94.72%	89.97%	92.46%	84.32%	90.55%	90.61%	87.14%	**75.06%**	**85.18%**
c5315	93.57%	96.68%	94.24%	97.81%	88.55%	90.29%	89.72%	91.12%	**87.06%**	**89.87%**
c7552	91.79%	95.66%	78.53%	99.72%	77.58%	95.34%	79.88%	94.29%	**71.43%**	**91.06%**
cavlc	93.20%	83.78%	96.83%	92.85%	94.28%	85.38%	**92.07%**	89.62%	94.35%	**81.18%**
priority	53.17%	97.16%	47.96%	98.77%	37.28%	98.54%	39.12%	97.22%	**32.25%**	**96.57%**
Average	85.65%	85.99%	77.41%	90.55%	73.60%	83.59%	74.69%	82.44%	**70.16%**	**76.76%**

[1] **Sanbao Su et al. (2022). "VECBEE: A versatile efficiency–accuracy configurable batch error estimation method for greedy approximate logic synthesis".** In: 41.11, pp. 5085–5099.

[2] **Chang Meng et al. (2023). "HEDALS: Highly Efficient Delay-driven Approximate Logic Synthesis".** In: 42.11, pp. 3491–3504.

Optimization Results under ED constraint

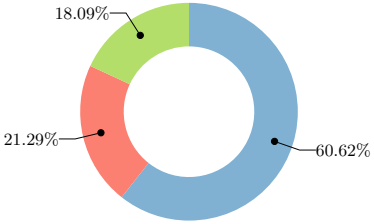
Table: Comparison of multi-objective optimization performance between our framework and other works under 1.96% *NMED* constraints.

Circuit	VECBEE-S		HEDALS		TCAD24[3]		DCGWO[4]		Ours	
	delay ratio	area ratio	delay ratio	area ratio	delay ratio	area ratio	delay ratio	area ratio	delay ratio	area ratio
c6288	97.33%	90.81%	73.28%	89.90%	74.38%	91.25%	76.92%	89.01%	**71.95%**	**87.72%**
adder	82.62%	96.42%	78.14%	94.02%	66.88%	92.96%	74.23%	93.69%	**59.23%**	**92.50%**
barshift	90.01%	83.87%	87.46%	89.98%	83.12%	80.55%	82.78%	83.26%	**82.16%**	**75.62%**
max	92.55%	86.44%	81.96%	93.24%	75.99%	85.64%	76.81%	91.80%	**74.86%**	**82.33%**
mult	95.89%	91.79%	82.59%	89.66%	78.99%	88.66%	80.08%	87.63%	**71.20%**	**86.98%**
sine	94.10%	90.28%	91.11%	93.48%	86.14%	88.56%	**82.10%**	89.78%	87.60%	**86.25%**
sqrt	83.23%	91.07%	75.03%	92.10%	75.66%	90.54%	79.16%	**86.29%**	**71.21%**	89.32%
square	92.53%	80.81%	82.13%	76.27%	79.36%	77.54%	82.58%	72.23%	**77.72%**	**71.86%**
Average	91.03%	88.94%	81.46%	89.83%	77.56%	86.95%	79.33%	86.46%	**74.49%**	**84.07%**

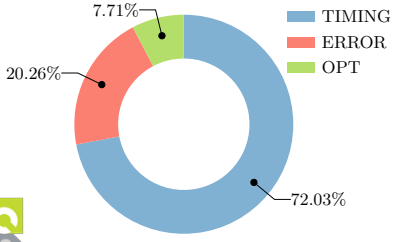
[3] **Yuyang Ye et al. (2024). "Timing-Driven Technology Mapping Approximation Based on Reinforcement Learning".** In.

[4] **Xiangfei Hu et al. (2024).** ***Timing-driven Approximate Logic Synthesis Based on Double-chase Grey Wolf Optimizer.*** arXiv: `2411.10990 [cs.AR]`. URL: `https://arxiv.org/abs/2411.10990`.

Runtime



(a) Runtime breakdown of DCGWO



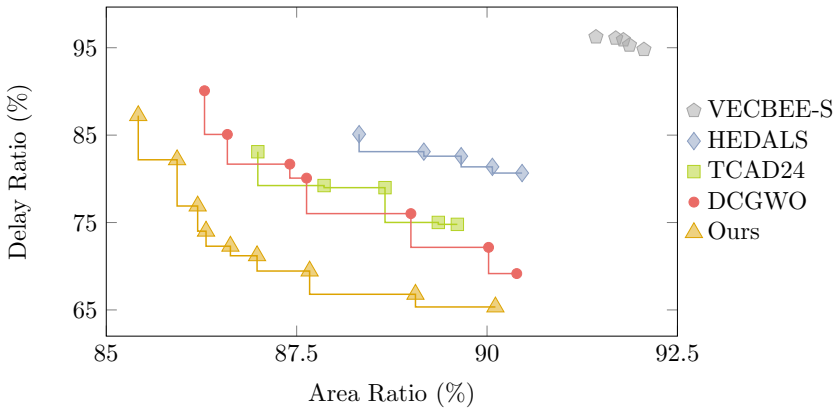
(b) Runtime breakdown of ours

Table: Overall Runtime (min.) Comparison.

Circuit	VECBEE-S	HEDALS	TCAD24	DCGWO	Ours
c6288	58.60	34.18	22.95	28.05	**20.32**
adder	22.63	18.17	17.68	15.92	**10.09**
barshift	37.89	30.11	22.69	21.26	**17.70**
max	34.68	39.44	23.31	27.13	**19.22**
mult	293.01	187.82	57.83	64.12	**40.38**
sine	71.88	51.63	51.24	42.35	**39.76**
sqrt	551.95	268.8	105.87	132.67	**98.95**
square	57.22	37.13	23.74	**20.95**	24.78
Average	140.98	83.41	40.66	44.06	**33.90**

Pareto Results

Multi-objective optimization results across all works on 128-bit multiplier under 1.96% NMED constraint.



Summary

62 THE CHIPS TO SYSTEMS CONFERENCE

SPONSORED BY CEDA IEEE Council on Electronic Design Automation sigda ACM

Summary

- We present the first multi-objective Approximate Logic Synthesis (ALS) framework implemented using Monte Carlo Tree Search (MCTS).
- We utilize non-dominated circuit ranking to guide MCTS in globally identifying approximate circuits with high optimization potential.
- We leverage the Rank-Transformer to predict path-domain rankings of local approximate changes (LACs) to select high-quality LACs within critical paths enable effective optimization of both delay and area.

AI

Security

Systems

EDA

Design

62 THE CHIPS TO SYSTEMS CONFERENCE

SPONSORED BY CEDA IEEE Council on Electronic Design Automation sigda