IEEE/ACM
2023 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN
ICCAD
42nd Edition

AlphaSyn: Logic Synthesis Optimization with Efficient Monte Carlo Tree Search

Zehua Pei[1], Fangzhou Liu[2], Zhuolun He[1], Guojin Chen[1],
Haisheng Zheng[2], Keren Zhu[1], **Bei Yu**[1]

[1] The Chinese University of Hong Kong
[2] Shanghai Artificial Intelligence Laboratory

Nov. 01, 2023

Outline

Introduction

Logic synthesis with Transformation Recipe

- **Transformation recipe** is a sequence of synthesis transformations, which is iteratively applied to the logic network.
- Optimality gap happens when the recipe is applied to different designs!
- Better transformation recipe enables better optimization!

```
alias resyn       "b; rw; rwz; b; rwz; b"
alias resyn2      "b; rw; rf; b; rw; rwz; b; rfz; rwz; b"
alias resyn2a     "b; rw; b; rw; rwz; b; rwz; b"
alias resyn3      "b; rs; rs -K 6; b; rsz; rsz -K 6; b; rsz -K 5; b"
alias compress    "b -l; rw -l; rwz -l; b -l; rwz -l; b -l"
alias compress2   "b -l; rw -l; rf -l; b -l; rw -l; rwz -l; b -l; rfz -l; rwz -l; b -l"
```

Recipe in ABC.[1]

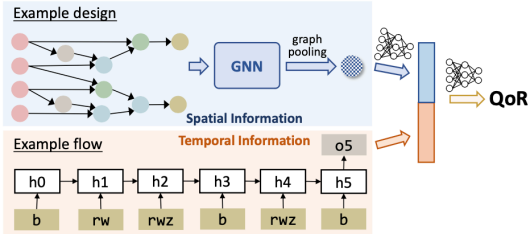
[1] https://github.com/berkeley-abc/abc/blob/master/abc.rc

Previous solutions

Classification & Prediction

Classify or predict the final Quality-of-Result (QoR) of synthesis sequences.[2]

- A large dataset is required for training and evaluation.
- The accuracy is limited and uncertain.



QoR prediction.[3]

[2]Cunxi Yu, Houping Xiao, and Giovanni De Micheli (2018). "Developing synthesis flows without human knowledge". In: *Proceedings of the 55th Annual Design Automation Conference*, pp. 1–6.

[3]Nan Wu et al. (2022). "Hybrid graph models for logic optimization via spatio-temporal information". In: *arXiv preprint arXiv:2201.08455*.

Previous solutions (cont'd)

Sequence Generation

Generate the sequence with specific optimization objectives by reinforcement learning (RL)[4], Bayesian optimization (BO)[5] or with heuristics.

- RL and BO based methods are **lack of enough exploration** by performing in a "forward" process, where the sequence is generated as trajectory and evaluated as a whole.
- The methods with heuristics always explore in the **reduced search space**, which results in local optimum.

[4]Keren Zhu et al. (2020). "Exploring logic optimizations with reinforcement learning and graph convolutional network". In: *Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD*, pp. 145–150.

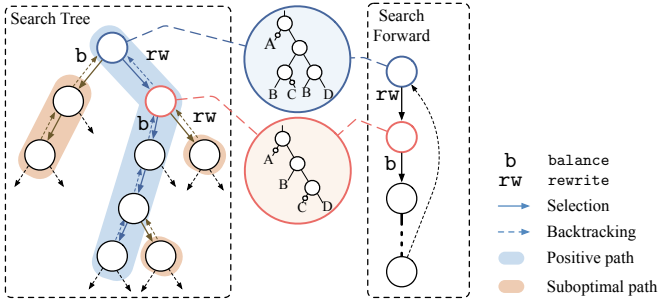
[5]Walter Lau Neto et al. (2022). "FlowTune: End-to-end Automatic Logic Optimization Exploration via Domain-specific Multi-armed Bandit". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

Our solution

Decision-making with search tree

Exploring the logic optimization sequences by constructing a search tree, and making decision with the tree statistics.

- The performance of actions in different stages are aware, and hence balancing the act between **pushing forward and going back**.
- The **statistics** summarized from the search tree is informative and convincing for the final decision.

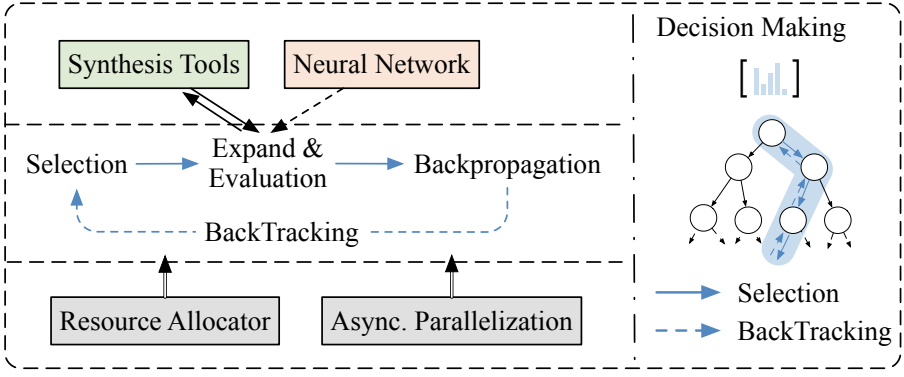


Search Tree and forward search.

Framework

Framework Overview

- Customized MCTS.
- Stable Learning Strategies.
- Acceleration for MCTS.



Overview of AlphaSyn.

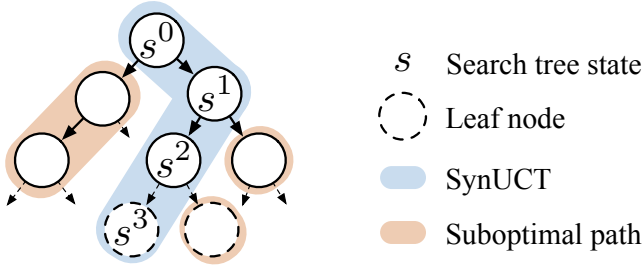
Selection

SynUCT:

$$a^t = \arg\max_{a \in \mathcal{A}} (Q^{t,a} + R^{t,a} + U^{t,a}). \tag{1}$$

$Q^{t,a}$: **long-term return**. $R^{t,a}$: **immediate reward**. $U^{t,a}$: **balance mechanism**:

$$U^{t,a} = c_{puct} \cdot P^{t,a} \cdot \frac{\sqrt{N^t}}{N^{t,a} + 1}. \tag{2}$$



Selection in the customized MCTS.

Expansion & Evaluation

Define the **reward** $R^{\mathcal{T}}$:

$$R^{\mathcal{T}} = \text{sgn}(O^{\mathcal{T}-1} - O^{\mathcal{T}}) \cdot \sqrt{\frac{|O^{\mathcal{T}-1} - O^{\mathcal{T}}|}{\textit{baseline}}}. \tag{3}$$

Multi-objecitves reward:

$$R^{\mathcal{T}} = (1-\beta) \cdot R_1^{\mathcal{T}} + \beta \cdot R_2^{\mathcal{T}}. \tag{4}$$

Then new nodes are **expanded** with initial statistics:

$$\{N^{\mathcal{T},a} = 0, Q^{\mathcal{T},a} = 0, P^{\mathcal{T},a} = p^{\mathcal{T},a}\}. \tag{5}$$

Backpropagation

Update the node visit count:

$$N^t \longleftarrow N^t + 1. \tag{6}$$

Update the long-term return Q^t, all $t < \mathcal{T}$:

$$Q^t \longleftarrow \lambda \cdot \max_{a \in \mathcal{A}} (Q^{t,a} + R^{t,a}). \tag{7}$$

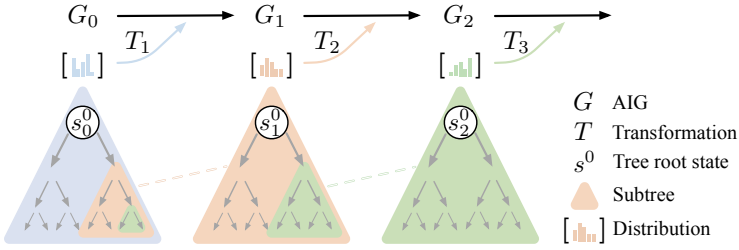
The **exploration-exploitation** trade-off:

- Accumulated Q prioritizes the exploration in the **shallow levels**.
- **Deeper exploitation** is also encouraged in promising directions.

Decision Making

Take action T_i depends on the statistics:

$$T_i = \arg\max_{a \in \mathcal{A}} (Q^{0,a} + R^{0,a}). \tag{8}$$



Decision making in AlphaSyn.

Data Collection and Training

Collecting data via *self-syn*, the accumulated Q and exponential distribution of (Q+R) are stored:

$$\pi^{0,a} = \frac{(Q^{0,a} + R^{0,a})^{1/\kappa}}{\sum_{b \in \mathcal{A}} (Q^{0,b} + R^{0,b})^{1/\kappa}}. \tag{9}$$

Input and Output of the model PQnet (f_θ):

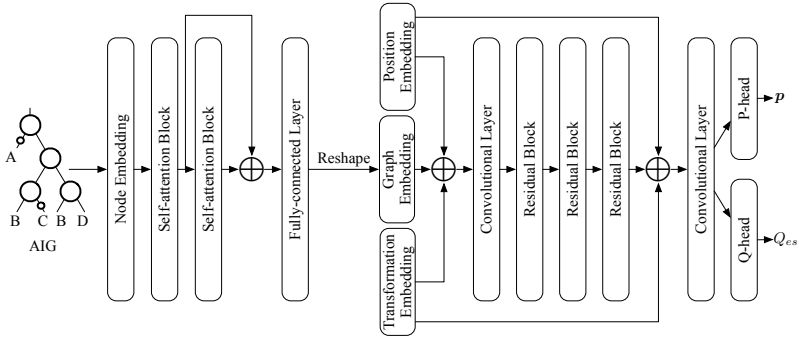
$$(\boldsymbol{p}^{\mathcal{T}}, Q_{es}^{\mathcal{T}}) = f_\theta(s_i^{\mathcal{T}},\ a_i^{\mathcal{T}-1},\ (\mathcal{T}-1)+i). \tag{10}$$

Loss function:

$$L_{\text{total}} = L_{\text{CE}}(\boldsymbol{p}^0, \boldsymbol{\pi}^0) + L_{\text{MSE}}(Q_{es}^0, Q^0). \tag{11}$$

Network Architecture

We design the PQnet based on SAGEConv[6] and the self-attention pooling SAGPool[7].



The overview of PQnet.

[6] Will Hamilton, Zhitao Ying, and Jure Leskovec (2017). "Inductive representation learning on large graphs". In: *Advances in neural information processing systems* 30.

[7] Junhyun Lee, Inyeop Lee, and Jaewoo Kang (2019). "Self-attention graph pooling". In: *International conference on machine learning*. PMLR, pp. 3734–3743.

Asynchronous Parallelization

Conditional **path-blocking-releasing** Asynchronous Parallelization, **Par-SynUCT**:

$$a^t_{par} = \arg\max_{a \in \mathcal{A}} (Q^{t,a} + R^{t,a} + U^{t,a} + B^{t,a}). \tag{12}$$

Based on conditions on the tree branch, Par-SynUCT acts differently:

- When no thread is selecting: tree branch is accessed.
- When a thread is selecting: tree branch is blocked and unaccessed.
- After selecting: tree branch is released.

Experiments

Logic Optimization

Table: Comparison with FlowTune[8] for logic optimization. Comparison with MLCAD'20[9] and ICCAD'21[10] for logic optimization.

	Flowtune		AlphaSyn w/o nn		AlphaSyn w/ nn		
Design	#N	rt (s)	#N̂	r̂t (s)	#N̂	#N	r̂t (s)
bfly	22740	495.81	22619.9	**223.60**	**22381**	**22582.7**	537.39
dscg	22258	482.35	22225.2	**210.20**	**22000**	**22155.9**	557.09
fir	21807	488.67	21611.6	**211.40**	**21544**	**21591.5**	543.77
ode	13038	260.16	12935.9	**112.98**	**12736**	**12768.7**	472.55
or1200	10316	118.80	10266.9	**74.16**	**10194**	**10195.2**	420.02
syn2	23633	504.19	23547.1	**242.00**	**23233**	**23535.7**	587.01
Average	18965.3	391.66	18867.8	**179.06**	**18689.5**	**18805.0**	519.64
Ratio	1.000	1.000	0.995	**0.457**	**0.985**	**0.992**	1.327

	9	10	AlphaSyn w/o nn		AlphaSyn w/ nn		
Design	#N	#N	#N̂	r̂t (s)	#N̂	#N	r̂t (s)
C1355	386.2	**386**	**386**	19.24	**386**	**386**	360.93
C6288	**1870**	**1870**	**1870**	24.35	**1870**	**1870**	454.89
C5315	1337.4	1315	1291.2	20.87	**1287**	**1289.4**	450.44
dalu	1039.8	1085	1008.4	21.22	**1007**	**1008.4**	475.94
k2	1128.4	1137	1045.9	22.66	**1035**	**1041.3**	451.69
mainpla	3438.4	3461	3406.3	24.81	**3386**	**3390.2**	462.66
apex1	1921.6	1885	1892	22.89	**1881**	**1883**	453.62
bc0	819.4	831	803.2	23.14	**795**	**798**	464.43
Average	1492.7	1496.3	1462.9	22.40	**1455.88**	**1458.3**	446.83
Ratio	1.000	1.002	0.980	-	**0.975**	**0.977**	-

[8] Walter Lau Neto et al. (2022). "FlowTune: End-to-end Automatic Logic Optimization Exploration via Domain-specific Multi-armed Bandit". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

[9] Keren Zhu et al. (2020). "Exploring logic optimizations with reinforcement learning and graph convolutional network". In: *Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD*, pp. 145–150.

[10] Yasasvi V Peruvemba et al. (2021). "RL-guided runtime-constrained heuristic exploration for logic synthesis". In: *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, pp. 1–9.

Standard-cell Technology Mapping

Table: Comparison with FlowTune[11] for technology mapping on Nangate 45nm library and ASAP 7nm library.

Design	Flowtune		AlphaSyn w/o nn		AlphaSyn w/ nn		
	Area (μm^2)	rt (s)	Area (μm^2)	rt (s)	Area (μm^2)	Area (μm^2)	rt (s)
bfly	16881.8	969.95	16089.0	**450.78**	**15493**	**15959.2**	830.20
dscg	16393.2	913.20	16142.2	**411.80**	**15743**	**16097.1**	958.01
fir	16323.5	922.73	15876.9	**431.70**	**15336**	**15724.4**	896.26
ode	9302.5	479.77	9138.4	**272.59**	**9101**	**9151.2**	710.97
or1200	6731.4	228.24	6756.6	**153.92**	**6689.6**	**6729.1**	531.53
syn2	17246.4	880.14	16410.6	**490.37**	**16051.5**	**16078.1**	943.32
Average	13813.1	732.34	13402.3	**368.53**	**13069.0**	**13289.9**	811.72
Ratio	1.000	1.000	0.970	**0.503**	**0.946**	**0.962**	1.108

Design	Flowtune		AlphaSyn w/o nn		AlphaSyn w/ nn		
	Area (μm^2)	rt (s)	Area (μm^2)	rt (s)	Area (μm^2)	Area (μm^2)	rt (s)
bfly	16934.2	2111.12	15586.7	**1396.17**	**14837.7**	**15059.7**	1520.04
dscg	15722.3	1865.30	15537.2	**1261.43**	**15031**	**15164**	1329.71
fir	16109	1784.02	16225.8	**1060.90**	**15301**	**15471**	1294.33
ode	9193.4	979.36	8319.8	**763.16**	**8123**	**8257.3**	919.07
or1200	**4107.6**	442.80	4241.9	**584.95**	4191.3	4201.1	719.72
syn2	18501.4	1812.36	15728.9	**1254.69**	**15217.3**	**15370.2**	1480.61
Average	13428	1499.16	12606.7	**1053.55**	**12116.9**	**12253.9**	1210.58
Ratio	1.000	1.000	0.939	**0.703**	**0.902**	**0.913**	0.808

[11] Walter Lau Neto et al. (2022). "FlowTune: End-to-end Automatic Logic Optimization Exploration via Domain-specific Multi-armed Bandit". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

FPGA Technology Mapping

Table: Comparison with FlowTune[12] , DRiLLS[13] and ASPDAC'23[14] for FPGA mapping.

	Flowtune		AlphaSyn w/o nn		AlphaSyn w/ nn		
Design	LUTs	rt (s)	LUTs	rt (s)	LUTs	LUTs	rt (s)
bfly	8187	1511.40	7974.6	**945.39**	**7889**	**7949.9**	1148.27
dscg	8191	1481.99	8018.9	**1005.74**	**7965**	**7972.9**	1065.17
fir	7958	1422.34	7815.6	**952.29**	**7728**	**7778.1**	1036.73
ode	5022	708.07	5010.7	**593.29**	**4973**	**4995.2**	830.44
or1200	2751	401.295	2725.2	**300.62**	**2714**	**2720.6**	564.46
syn2	8316	1510.25	8215.4	**968.38**	**8172**	**8189.7**	1142.71
Average	6737.5	1172.56	6626.7	**794.27**	**6573.5**	**6601.1**	964.63
Ratio	1.000	1.000	0.984	**0.677**	**0.976**	**0.980**	0.823

	13	14	AlphaSyn w/o nn		AlphaSyn w/ nn		
Design	LUTs	LUTs	LUTs	rt (s)	LUTs	LUTs	rt (s)
max	694	687.8	680.5	74.59	**674**	**680**	342.56
adder	**244**	**244**	**244**	62.74	**244**	**244**	368.74
cavlc	112.2	111.3	106.8	53.14	**106**	**106**	321.12
ctrl	**28**	**28**	**28**	38.81	**28**	**28**	341
int2float	42.6	42.3	39.2	56.62	**39**	**39**	332.82
router	70.1	69.5	65.6	24.59	**65**	**65**	320.43
priority	133.4	142.9	135.6	59.15	**131**	135	350.11
i2c	292.1	289.32	280.6	47.78	**272**	**280**	373.46
sin	1441.5	1438	1439.7	91.02	**1435**	**1438**	406.19
square	3889.4	3889	3877	166.27	**3875**	**3877**	523.26
sqrt	4708	4685.3	**4415**	269.59	**4415**	**4415**	589.93
log2	7583.6	7580.1	**7580**	365.77	**7580**	**7580**	706.96
multiplier	5678	5672	5687.5	245.75	**5670.5**	**5672**	620.53
voter	1834.7	1678.1	1538.8	111.44	**1534**	**1537.4**	470.64
div	7944.4	7807.1	6685.3	244.04	**5088.4**	**6650.1**	712.57
mem_ctrl	10527.6	10309.7	9567.7	71.63	**9211.5**	**9513.2**	659.67
Average	2826.5	2792.2	2648.2	123.93	**2523.0**	**2641.2**	464.99
Ratio	1.000	0.988	0.937	-	**0.893**	**0.934**	-

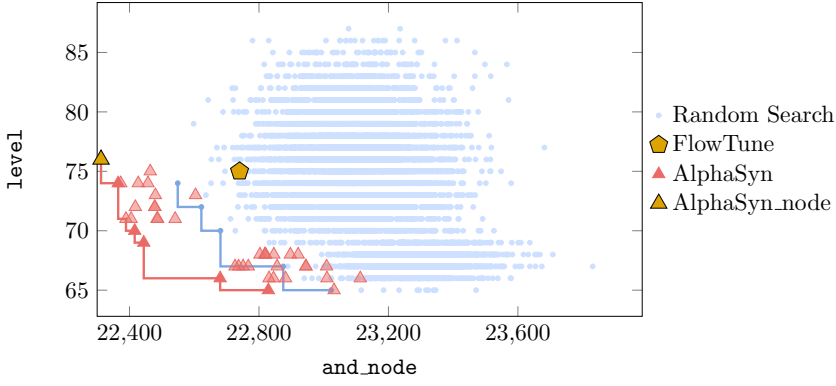
[12]Walter Lau Neto et al. (2022). "FlowTune: End-to-end Automatic Logic Optimization Exploration via Domain-specific Multi-armed Bandit". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

[13]Abdelrahman Hosny et al. (2020). "DRiLLS: Deep reinforcement learning for logic synthesis". In: *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, pp. 581–586.

[14]Guanglei Zhou and Jason H Anderson (2023). "Area-Driven FPGA Logic Synthesis Using Reinforcement Learning". In: *Proceedings of the 28th Asia and South Pacific Design Automation*

Multi-objective Synthesis

Experienment is also conducted for **Multi-objective Synthesis**. Bule points are 100000 randomly generated sequences.



By tuning the weight between `and_node` and `level`, a Pareto front has been formed by AlphaSyn's results. The result on objective of pure `and_node` minimization is specially marked ("AlphaSyn_node") for fair comparison with FlowTune.

THANK YOU!