



DESIGN, AUTOMATION
AND TEST IN EUROPE

THE EUROPEAN EVENT FOR
ELECTRONIC SYSTEM DESIGN & TEST

31 MARCH – 2 APRIL 2025
LYON, FRANCE

CENTRE DE CONGRÈS DE LYON



Timing-driven Approximate Logic Synthesis Based on Double-chase Grey Wolf Optimizer

Xiangfei Hu¹, Yuyang Ye², Tinghuan Chen³, Hao Yan¹, Bei Yu²

¹Southeast University

²The Chinese University of Hong Kong

³The Chinese University of Hong Kong, Shenzhen

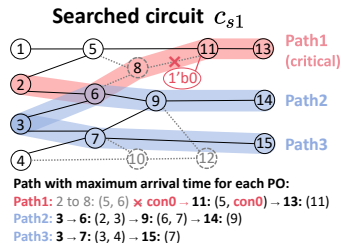
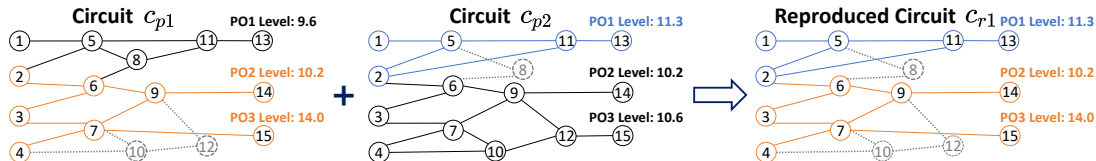
April. 02 , 2025



Approximate Logic Synthesis (ALS)

ALS can generate well-optimized circuits under given error constraints by applying **Local Approximate Changes (LACs)** on circuits.

- **Problem 1:** Focusing solely on critical path depth shortening or area minimization undermines comprehensive timing optimization.
- **Solution 1 (Timing-driven ALS):** Optimizing both of them & convert reduced area into **enhanced gate drive strength** under area constraints.
- **Problem 2:** Inadequate trade-off between objectives leads to local optima.
- **Solution 2 (Double-chase Grey Wolf Optimizer):** guide **finely partitioned** circuits population for global search and local convergence by **approximate actions**.



- Circuit searching applies LACs within each critical path. These LACs are filtered based on **output similarity**.
- Circuit reproduction can aggregate effective LACs from high-quality circuits via **PO-TFI evaluation function**:

$$Level(PO_i) = w_t \times \frac{1}{T_a(PO_i)} + w_e \times \frac{1}{Error(PO_i)}, \quad (1)$$

Double-chase consists of two stages:

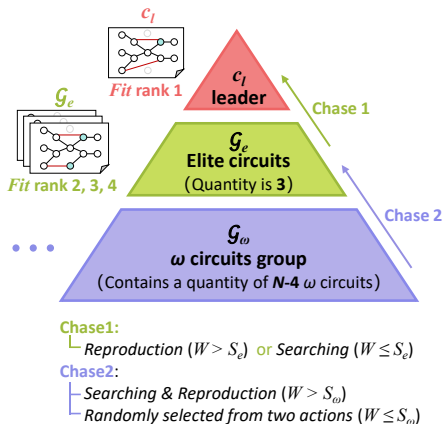
- Population partition based on fitness of circuits:

$$Fit = w_d \times \frac{Depth_{ori}}{Depth_{app}} + w_a \times \frac{Area_{ori}}{Area_{app}}. \quad (2)$$

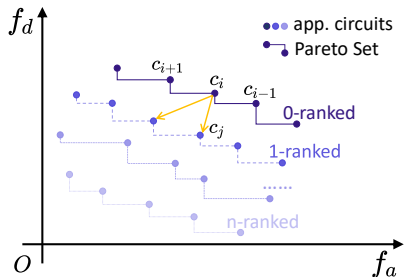
- Chase based on relationships between decision parameter W and the threshold S . W provides a dynamic correction to fitness distance D .

$$D = \begin{cases} r_c \times Fit(c_l) - Fit(c_i) & \forall c_i \in \mathcal{G}_e \\ \frac{r_c}{3} \sum_{c_j \in \mathcal{G}_e} Fit(c_j) - Fit(c_i) & \forall c_i \in \mathcal{G}_\omega \end{cases}, \quad (3)$$

$$W(c_i) = A \times D(c_i). \quad (4)$$



The Population Update selects high-quality circuits using **non-dominated sorting**.



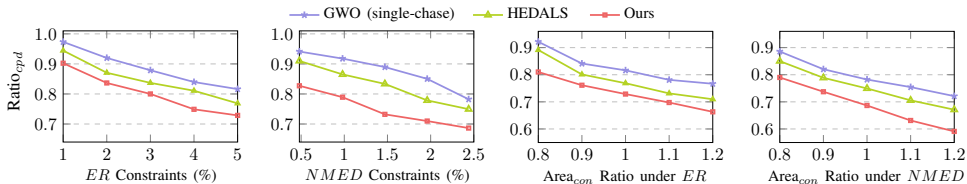
- Pareto Set Partition: categorizes the population of circuits without error violations into sets of different ranks using **Pareto dominance**.
- Crowding Distance Calculation: further sorts circuits within each Pareto set using *Dist*.

$$Dist(c_i) = \sum_{x=d,a} \frac{f_x(c_{i-1}) - f_x(c_{i+1})}{\max_k(f_x) - \min_k(f_x)}. \quad (5)$$

For the Pareto-optimal circuits, post-optimization removes its dangling gates and **resizes its remaining gates** without adjusting any circuit structure under $Area_{con}$.

Table: Comparison of performance between all works under 2.44% *NMED* constraints.

Circuit	Area _{con} (μm^2)	VECBEE-S ¹		Genetic ²		HEDALS ³		GWO (traditional)		Ours	
		Ratio _{cpd}	runtime(s)	Ratio _{cpd}	runtime(s)	Ratio _{cpd}	runtime(s)	Ratio _{cpd}	runtime(s)	Ratio _{cpd}	runtime(s)
int2float	194.00	0.9331	71.23	0.5047	151.73	0.7649	32.68	0.6010	178.30	0.4496	132.12
c6288	687.00	0.9663	4410.29	0.8696	3279.62	0.6368	2563.41	0.9079	2991.00	0.8313	2103.88
adder	495.00	0.7814	1697.37	0.8133	2083.15	0.7110	1362.70	0.8008	1550.03	0.6917	1193.71
barshift	1806.00	0.8670	2005.14	0.8287	2919.21	0.8025	1370.46	0.8166	1937.60	0.7271	1200.58
max	954.00	0.8809	2600.78	0.8933	3397.50	0.8355	2992.08	0.7517	3121.44	0.6799	2035.62
mult	31635.0	0.9010	17230.16	0.7818	12298.11	0.7068	9677.43	0.7276	9071.60	0.6459	6283.76
sine	4367.00	0.9187	5391.68	0.8326	3872.31	0.7945	3380.52	0.8722	4392.77	0.7603	3176.46
sqrt	6262.00	0.7993	33117.12	0.8011	20160.76	0.7437	11242.29	0.7803	17894.50	0.7058	9950.11
Average	5800.00	0.8809	8315.47	0.7906	6020.30	0.7494	4077.69	0.7823	5142.16	0.6865	3259.53



¹S. Su et al. (2022), "VECBEE: A versatile efficiency-accuracy configurable batch error estimation method for greedy ALS," *IEEE TCAD*.

²K. Balaskas et al. (2022), "Variability-aware approximate circuit synthesis via genetic optimization," *IEEE TCAS I*.

³C. Meng et al. (2023), "Hedals: Highly efficient delay-driven approximate logic synthesis," *IEEE TCAD*.