

# ASP-DAC 2024

29th Asia and South Pacific Design Automation Conference

Date: Jan. 22-25, 2024



## E2E-Check: End to End GPU-Accelerated Design Rule Checking with Novel Mask Boolean Algorithms

**Yifei Zhou**, Zijian Wang, Chao Wang

Southeast University

Jan. 24, 2024



ASIA SOUTH PACIFIC

DAAC

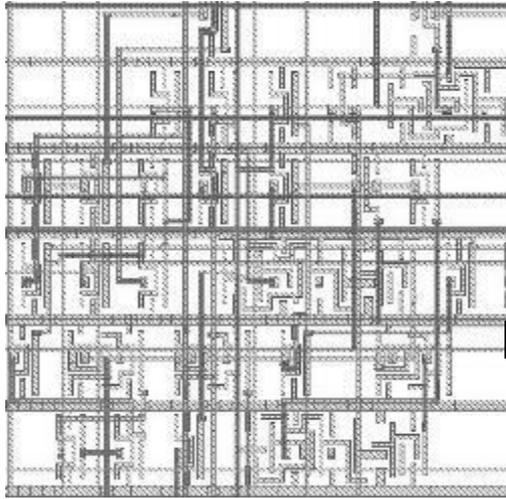
DESIGN

AUTOMATION

CONFERENCE

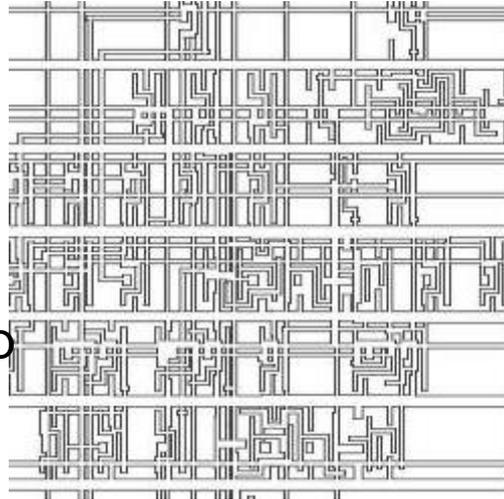
## **Background and Motivation**

- With the continues scaling-down of feature size, DRC rules are increasing, the massive number of polygons in layout continues to grow. The process is very time-consuming.
- Distance type rules, account for a significant proportion of all DRC rules.
- Merge operation on layout is needed before DRC checking, which is time-consuming and memory-consuming.
- It is easy to check distance type rule of an edge pair, but there are a massive number of edge pairs.



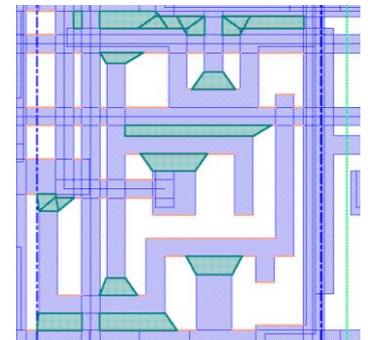
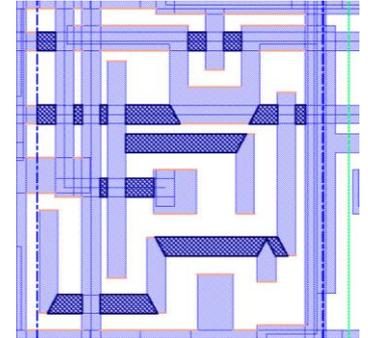
Layout

➔  
Merge op



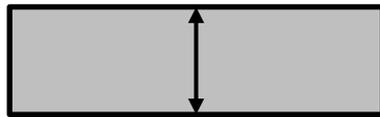
Merged  
Layout

➔  
Check  
&  
Report

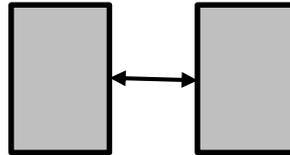


Violations

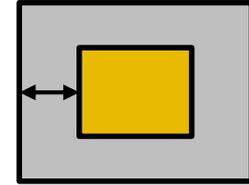
## ■ What information is needed for these DRC checks



Width

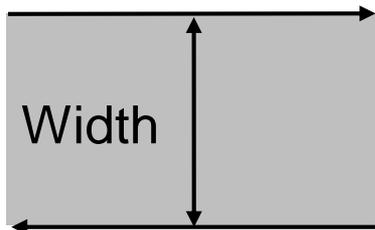
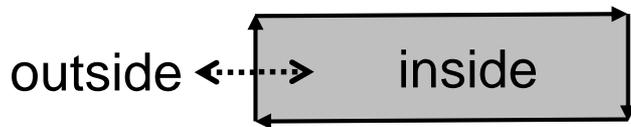


Space

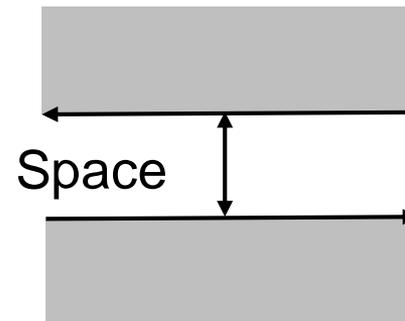


Enclosure

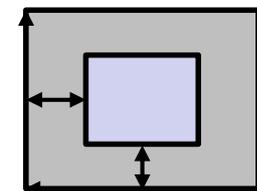
- Directed edges can determine the inside and outside of a polygon.
- Only the contour edges of MERGED layer need to be check.
- Distinguish different layers.



Width



Space



Enclosure

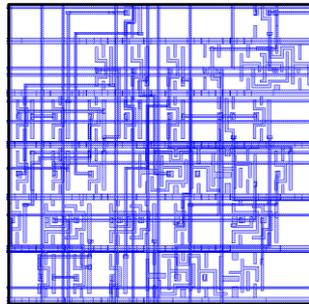
- There is a lot of repetitive work, i.e. edge pairs check, which is very suitable for GPUs to complete.
- Some tasks require a lot of logical operation, i.e. geometry Boolean operation, with a large number of intermediate states and significant overhead in large-scale data.
  - Directed contour edges are needed, and connection information is not required.
- Some jobs can be avoided and they consume a lot of resources.

ASIA SOUTH PACIFIC

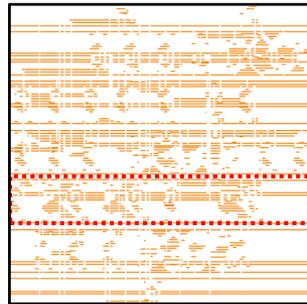
**DAC** DESIGN  
AUTOMATION  
CONFERENCE

**E2E-Check**

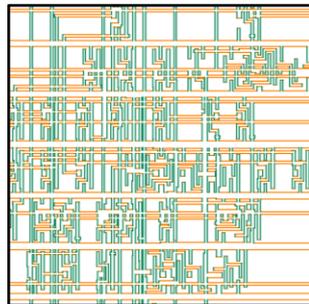
# An Overview of Our Proposed Solution



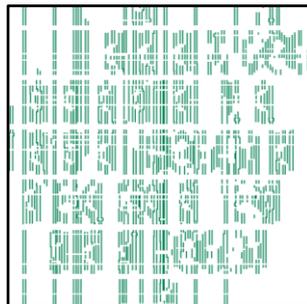
Layout



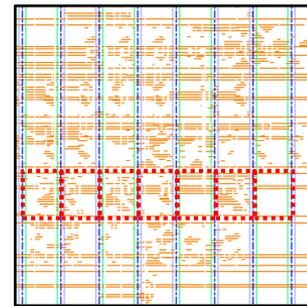
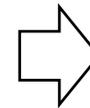
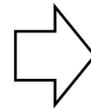
Horizontal  
Candidate  
edges



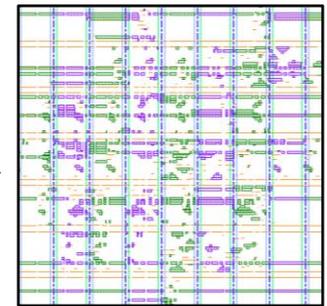
Candidate  
edges



Vertical  
Candidate  
edges



Heuristic  
Pruning



Checks  
and Report

## Manhattan polygon modeling

$$f_p(x, y) = \begin{array}{|c|c|} \hline & 0 \\ \hline 1 & \\ \hline \end{array}$$

(a)

$$f_\sigma(x, y) = \frac{d}{dy} \begin{array}{|c|c|} \hline & 0 \\ \hline 1 & \\ \hline \end{array} = \begin{array}{|c|c|} \hline \overline{-1} & \\ \hline +1 & \underline{-1} \\ \hline \end{array}$$

(b)

$$f_c(x, y) = \begin{array}{|c|c|c|} \hline & 1 & 0 \\ \hline 1 & 2 & 1 \\ \hline & 1 & \\ \hline \end{array}$$

(c)

$$\delta(x, y) = \frac{d}{dx} \frac{d}{dy} \begin{array}{|c|c|} \hline & 0 \\ \hline 1 & \\ \hline \end{array} = \begin{array}{|c|c|} \hline \dot{-1} & \dot{+1} \\ \hline & \dot{-1} & +1 \cdot \\ \hline \cdot +1 & & -1 \cdot \\ \hline \end{array}$$

(d)

- For clockwise direction, directed edges can be denoted as

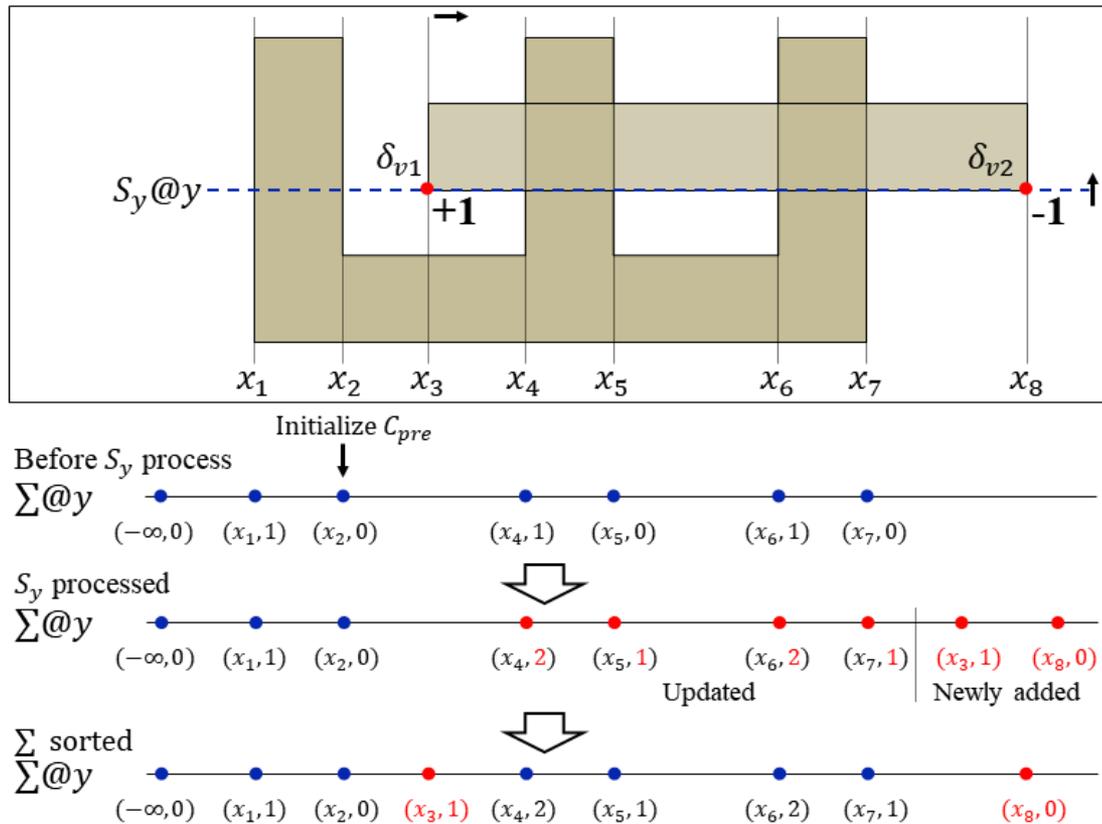
$$\vec{e} = (\hat{\delta}_s, \hat{\delta}_e) \quad \hat{\delta}_s = (p_s, -1) \quad \hat{\delta}_e = (p_e, +1)$$

- The Manhattan polygon can be represented by Augmented Vertices  $\hat{\delta}$ .

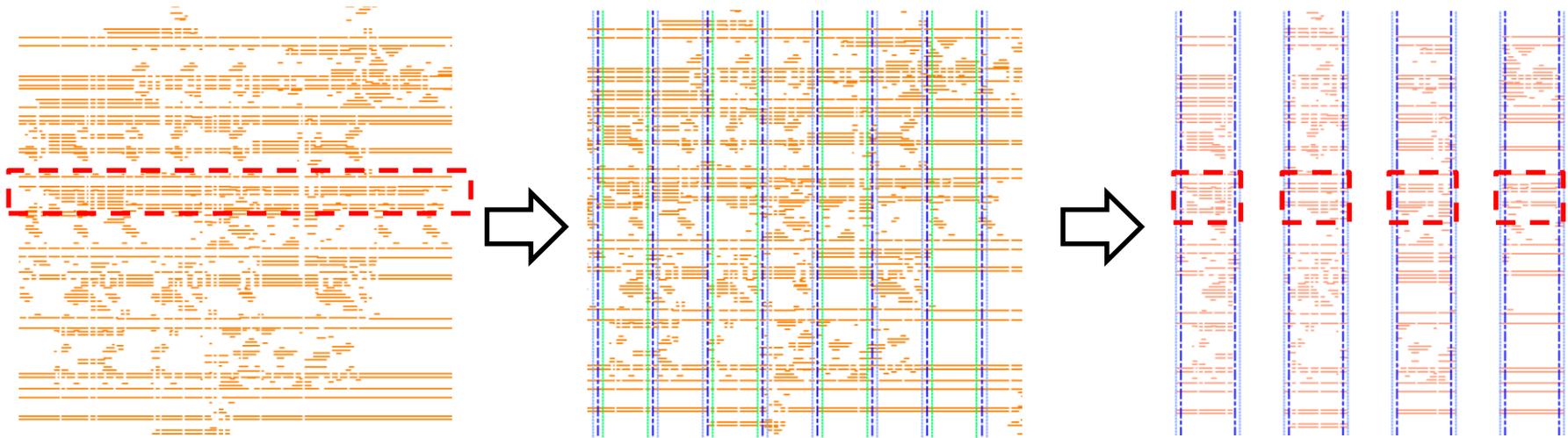


# The Proposed FCEC Algorithm

- Decompose the **merge** task into **sort** + **sweep** integration
  - The **sort** part is widely researched and can be parallelized
  - The **sweep** part:



- Reduce redundant computing and data transmission
  - FCEC algorithm outputs ordered CEs
  - Total computation cost can be estimated before checking
  - No real slice during the estimation phase
  - $O(N)$  time complexity of each round of detection



ASIA SOUTH PACIFIC

DAAC DESIGN  
AUTOMATION  
CONFERENCE

## Experimental Results

Table: Runtime Comparisons of Single-Thread Merge

Design	Layer	FCEC Time(s)	KLayout		Polygon90 [15]	
			Time(s)	FCEC Speedup	Time(s)	FCEC Speedup
gcd	M1	0.005	0.029	5.80×	0.014	2.80×
aes	M1	0.559	12.437	22.25×	1.592	2.85×
bp_be	M1	7.754	2283.973	294.45×	17.578	2.27×
bp	M1	20.179	8674.928	429.85×	38.177	1.89×

Table: Memory Comparisons of Single-Thread Merge

Design	Layer	FCEC	KLayout		Polygon90	
		VmSize(MB)	VmSize(MB)	Reduction by FCEC	VmSize(MB)	Reduction by FCEC
gcd	M1	4	4	-	4	-
aes	M1	64	437	85.4%	148	56.8%
bp_be	M1	674	9549	93.0%	3237	79.2%
bp	M1	1046	19197	94.6%	6089	82.8%

---

Lucanus J. Simonson(2010), "Industrial strength polygon clipping: A novel algorithm with applications in VLSI CAD," *Comput. Aided Des*, vol. 42, no. 12, pp. 1189–1196.

# The Experimental Results

## Heuristic Pruning @ Different Tiling Number

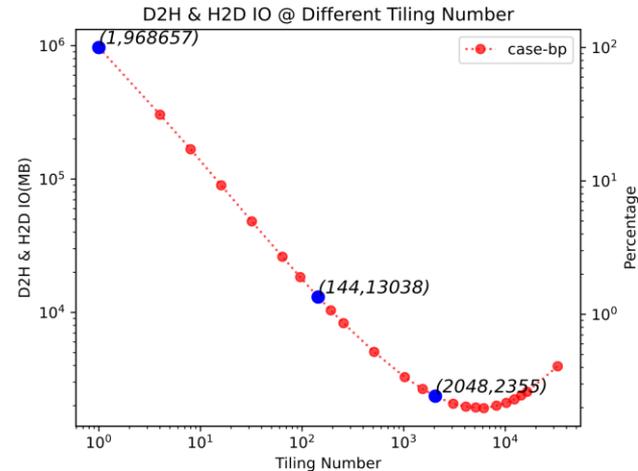
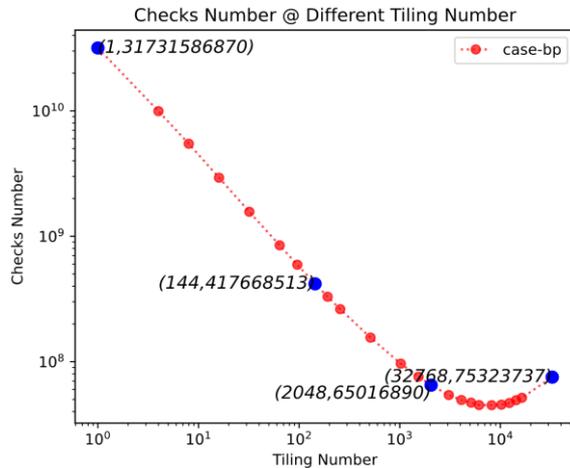
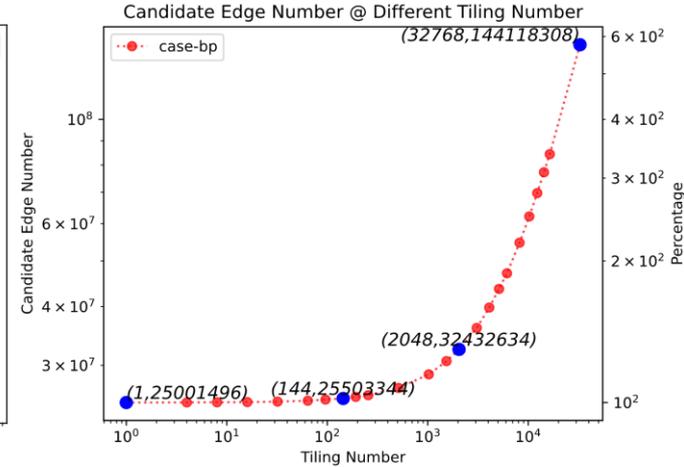
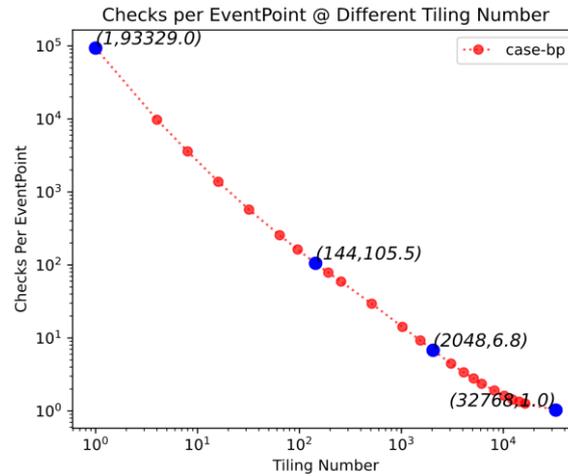
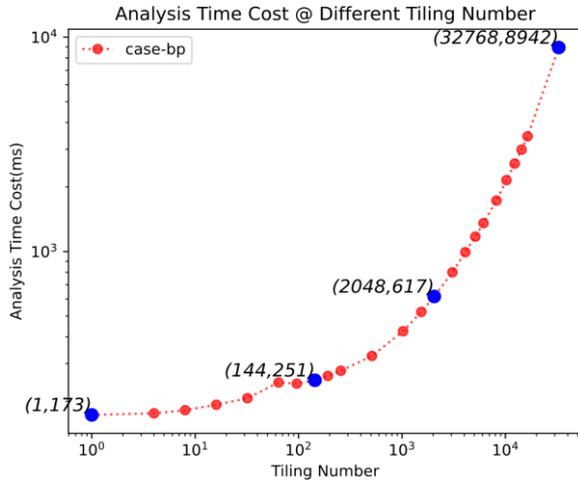


Table: Runtime Comparisons of Spacing Check and Enclosure Check

Design	Layer	Spacing Check (s)				Enclosure Check (s)			
		KLayout	E2E-Check	X-Check Speedup	E2E-Check Speedup	KLayout	E2E-Check	X-Check Speedup	E2E-Check Speedup
gcd	M1	5.2	0.10	5.25×	52.00×	23.2	0.10	16.00×	232.00×
aes	M1	282	0.97	66.57×	290.72×	1332	0.89	1257.76×	1496.63×
bp_be	M1	5312	8.4	54.31×	632.38×	38451	7.5	514.73×	5126.80×
bp	M1	12809	16.7	60.07×	767.00×	78284	14.5	418.06×	5398.89×
Average					443.03×				3063.50×

---

Z. He, Y. Ma, and B. Yu(2022), “X-Check: GPU-Accelerated Design Rule Checking via Parallel SweepLine Algorithms,” In Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design (ICCAD '22), no. 52, pp. 1–9

**THANK YOU!**

