DATE 23
DESIGN, AUTOMATION & TEST IN EUROPE
17 – 19 April 2023 · Antwerp, Belgium
The European Event for Electronic System Design & Test

Efficient Design Rule Checking with GPU Acceleration

Wei Zhong[1], **Zhenhua Feng**[1], Zhuolun He[2], Weimin Wang[1], Yuzhe Ma[3], Bei Yu[2]

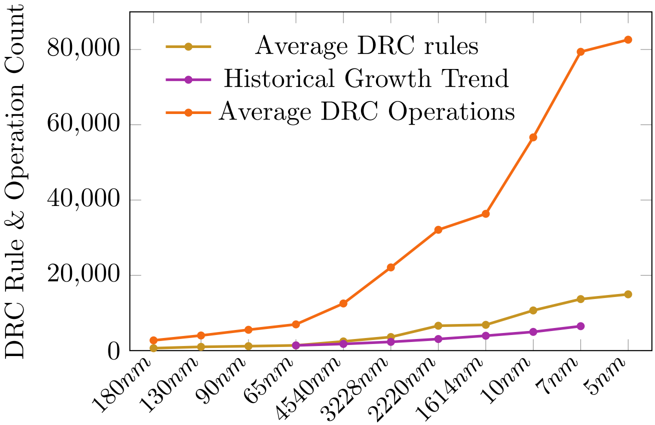
[1]Dalian University of Technology
[2]Chinese University of Hong Kong
[3]HKUST(GZ)

Apr. 18, 2023

Introduction

Design Rule Checking: Basic Concepts

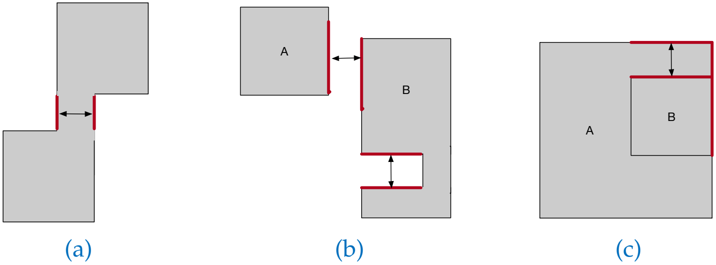
- Design rule checking (DRC) is the process to verify that a design layout conforms to a set of predefined design rules.



Growth of design rules.

Basic Design rules

- Width check
- Space check
- Enclosing check

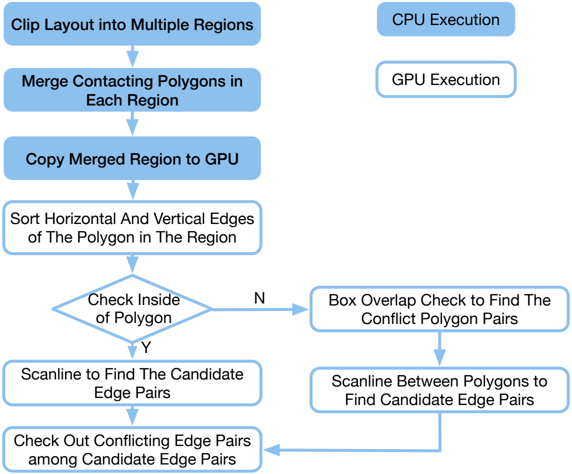


(a) Example of width check; (b) Example of space check; (c) Example of enclosing check.

Algorithms

Algorithm

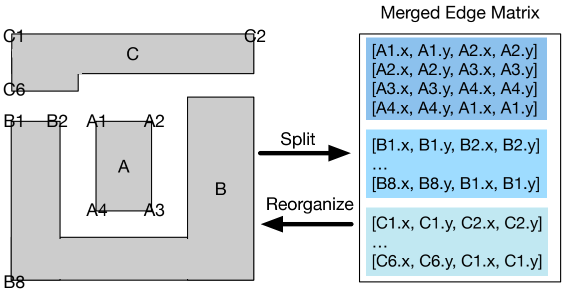
- The overview of our GPU-accelerated design rule checking flow.



Overview of our GPU-accelerate DRC.

Data Transfer

- The method of transferring layout information from the CPU to the GPU.

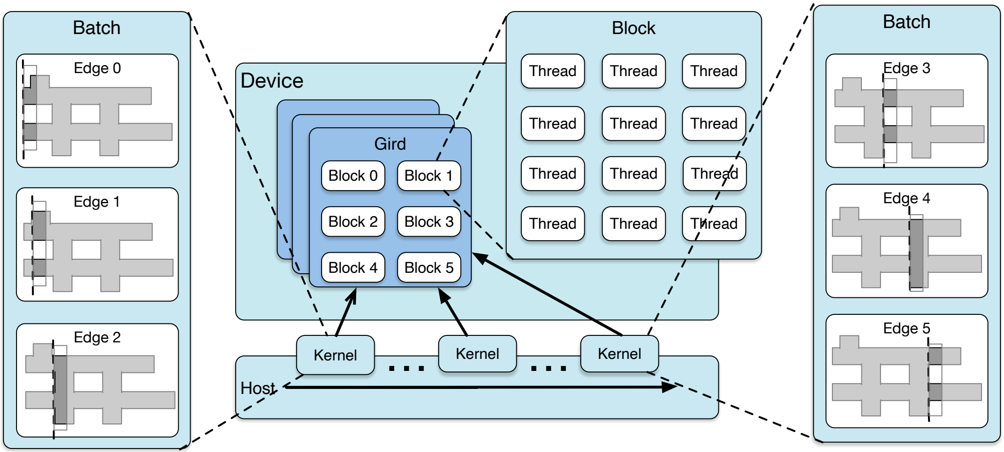


Encoding and decoding of a layout. Polygon edges are separated and stored uniformly in an edge matrix. The compact representation reduces data transfer overhead.

GPU-based Scanline

- Scanline inside polygon.
- Overlap checking.
- Scanline Between Polygons.

Scanline inside polygon



Programming architecture for the scanline algorithm.

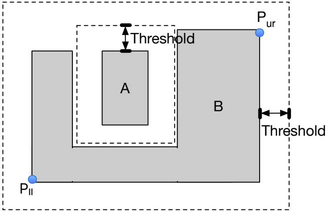
Scanline inside polygon

Suppose there are two vertical edges $e_1 = x_1 \times [y_{11}, y_{12}]$ and $e_2 = x_2 \times [y_{21}, y_{22}]$. If the their distance is smaller than the threshold ϵ:

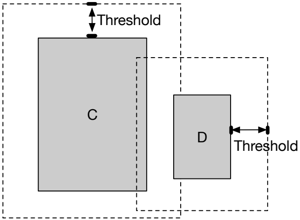
$$|x_1 - x_2| < \epsilon, \tag{1}$$

they will be recorded as an candidate edge pair.
Similarly, for horizontal $e_3 = [x_{31}, x_{32}] \times y_3$ and $e_4 = [x_{41}, x_{42}] \times y_4$, they will be recorded when the condition $|y_3 - y_4| < \epsilon$. holds.

Overlap checking



(a)



(b)

Two typical overlapping examples.

For any two polygons $P^{(i)}$ and $P^{(j)}$, threshold ϵ, if any one of the following formulas is satisfied, they will be regarded as a pair of polygons with the possibility of violation:

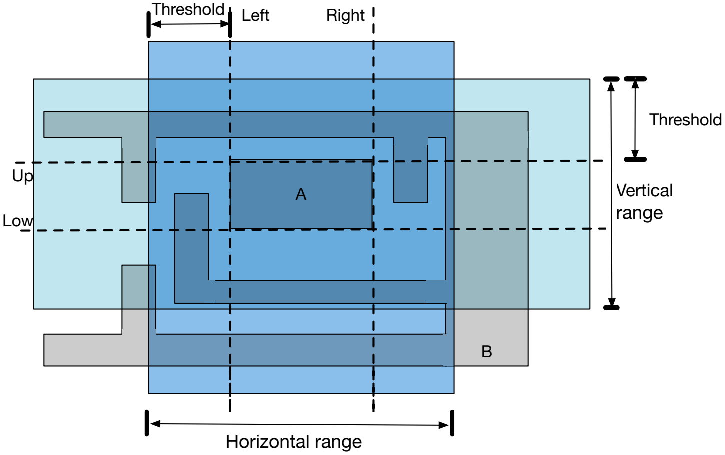
$$P_{ll}^{(i)}.x < P_{ur}^{(j)}.x + \epsilon, \tag{2}$$

$$P_{ll}^{(j)}.x < P_{ur}^{(i)}.x + \epsilon, \tag{3}$$

$$P_{ll}^{(i)}.x < P_{ur}^{(j)}.y + \epsilon, \tag{4}$$

$$P_{ll}^{(j)}.x < P_{ur}^{(i)}.y + \epsilon, \tag{5}$$

Enclosing check



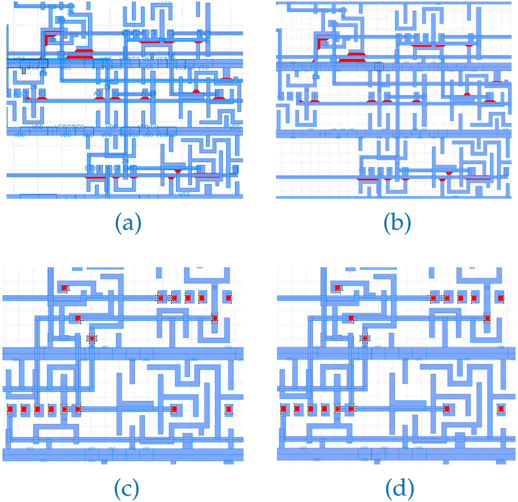
Polygon A and Polygon B are detected using the scanline algorithm between polygons.

Experimental result

Experimental setup

- A 64-bit Ubuntu Linux machine with TITAN RTX GPU and 3.5GHz Intel Core i9-10920X CPU.
- The compilers include CUDA NVCC 10.2 and GNU GCC 5.4.0.
- 4096 threads for all kernel configurations and 1 CPU core for all host operations.
- Baseline is KLayout 0.26.6.
- The layout benchmarks in the experiments are generated by OpenROAD project with default settings.

Correctness of DRC Results



(a) (b) (c) (d)

Space check results with (a) DRC script and (b) ours; Enclosing check results with (c) DRC script and (D) ours.

Runtime Performance

Table: Enclosing check in `Metal1`

Design	gcd	aes	bp_be	bp
8 CPU threads	33.522	13194.039	58477.239	90250.85
16 CPU threads	34.212	13074.176	51671.131	85792.708
24 CPU threads	34.52	13072.36	49047.536	74497.754
Ours	**0.343**	**27.932**	**257.056**	**409.381**
Speedup	**100.641**$\times$	**468.01**$\times$	**190.80**$\times$	**181.98**$\times$
Average	**201.1**$\times$			

Runtime Performance

Table: Enclosing check in `Metal2`

Design	gcd	aes	bp_be	bp
8 CPU threads	5.547	1977.047	2859.979	3332.67
16 CPU threads	5.732	1997.85	2435.594	2321.697
24 CPU threads	5.552	1976.503	2320.845	2298.961
Ours	**0.291**	**30.493**	**132.717**	**250.022**
Speedup	**19.08**×	**64.82**×	**17.49**×	**11.21**×
Average	**22.19**×			

Runtime Performance

Table: Space check in `Metal1`

Design	gcd	aes	bp_be	bp
8 CPU threads	10.99	376.244	5950.007	14865.705
16 CPU threads	11.131	3692.87	4540.09	8833.2
24 CPU threads	10.989	3690.08	4226.62	7565.84
Ours	**0.316**	**19.091**	**250.799**	**471.71**
Speedup	**34.78**×	**193.29**×	**16.85**×	**16.04**×
Average	**36.71**×			

Runtime Performance

Table: Space check in `Metal2`

Design	gcd	aes	bp_be	bp
8 CPU threads	6.378	2732.5534	3870.166	5015.233
16 CPU threads	6.168	2703.47	3365.211	3767.176
24 CPU threads	6.174	2666.539	3114.918	3621.914
Ours	**0.399**	**28.591**	**121.279**	**238.966**
Speedup	**15.47**×	**93.26**×	**25.68**×	**15.16**×
Average	**27.38**×			

Conclusion

GPU-based Scanline

- An efficient data packing method customized for layout data transfer.
- A GPU-based scanline algorithm that can perform parallel scanning of complex graphics.
- Speedup of 36× when performing space check on the `Metal1` layer.
- Speedup of 201× when performing enclosing check of the `Metal1` layer.

THANK YOU!