



超大规模版图图形匹配算法 赛题分析

杨 帆

复旦大学

专用集成电路与系统国家重点实验室

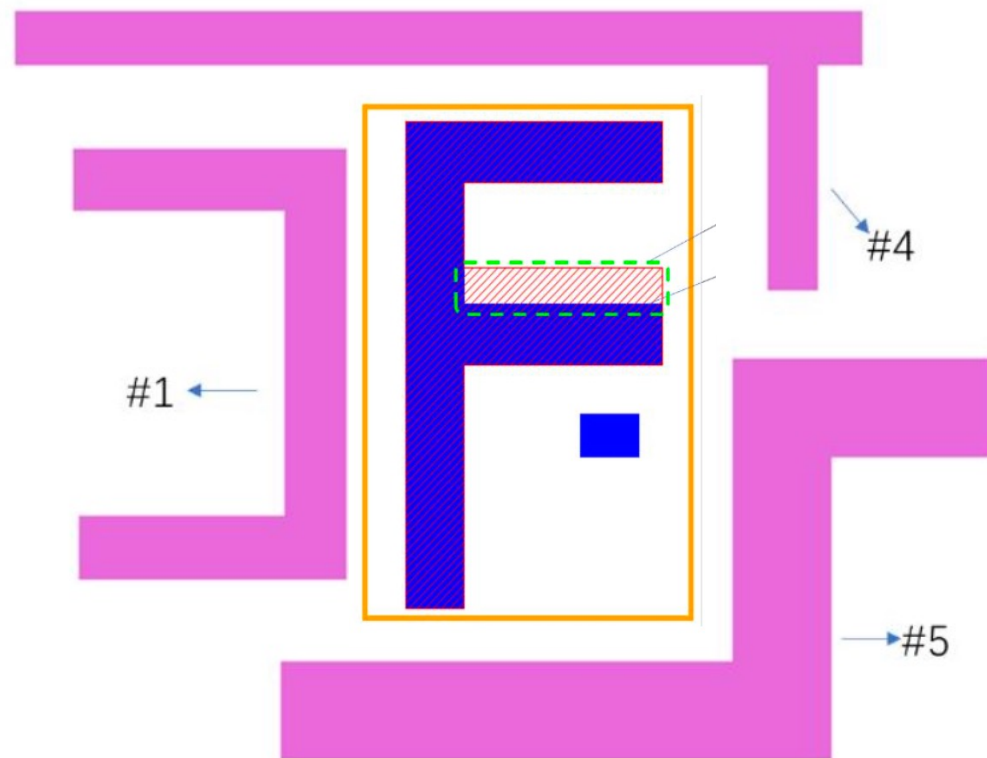
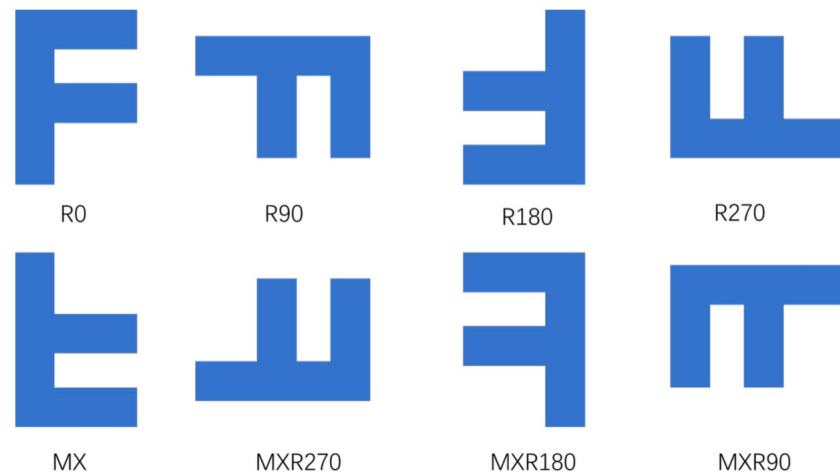
2023年9月15日

赛题背景

- 芯片制造过程的缺陷是影响芯片良率的重要因素，根据成因通常可分为系统性缺陷和非系统性缺陷。其中系统性缺陷通常由光刻，蚀刻，化学机械研磨等工艺引入，且与版图密切相关。
- 要想在版图中定位这些热点图形的位置以及在后续的流片中规避此类热点图形，就需要一种图形匹配(Pattern Matching)工具能将特定结构的图形在规模日益庞大的版图中定位出来。
- 传统的 DRC(Design Rule Check)工具仅能实现简单结构的图形搜索匹配，无法描述复杂图形。
- 图形匹配技术已被认为是在光刻工艺热点最准确且快速的检测手段，除此之外，图形匹配技术还被广泛用于标准单元图形检查，图形自动修复替换，重复单元图形加速 OPC 修正等多个领域。

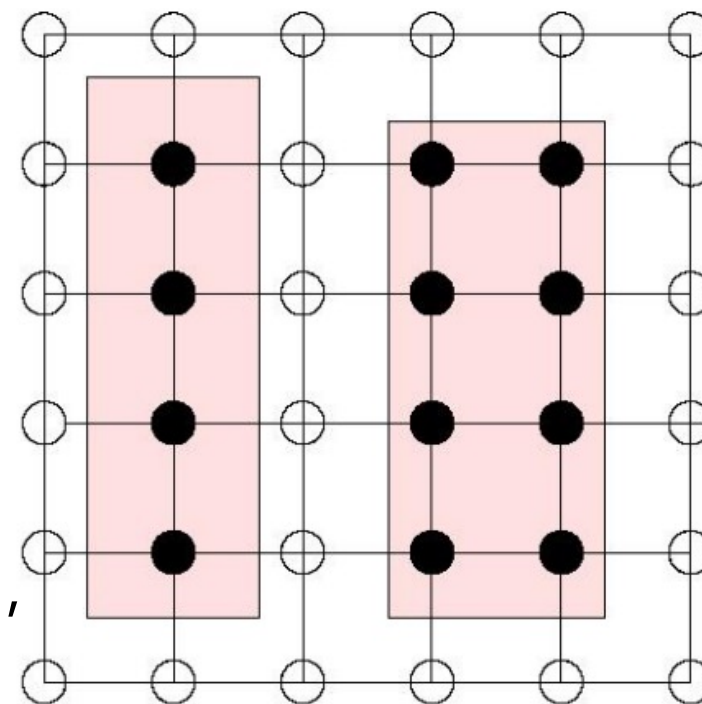
赛题指南

- 热点图形在大规模版图上的模糊匹配；
- 会给出模版图形和待匹配的版图，需要选手给出在版图上匹配的区域位置；
- 还会给出模版图形发生旋转和镜像情况下的匹配结果。



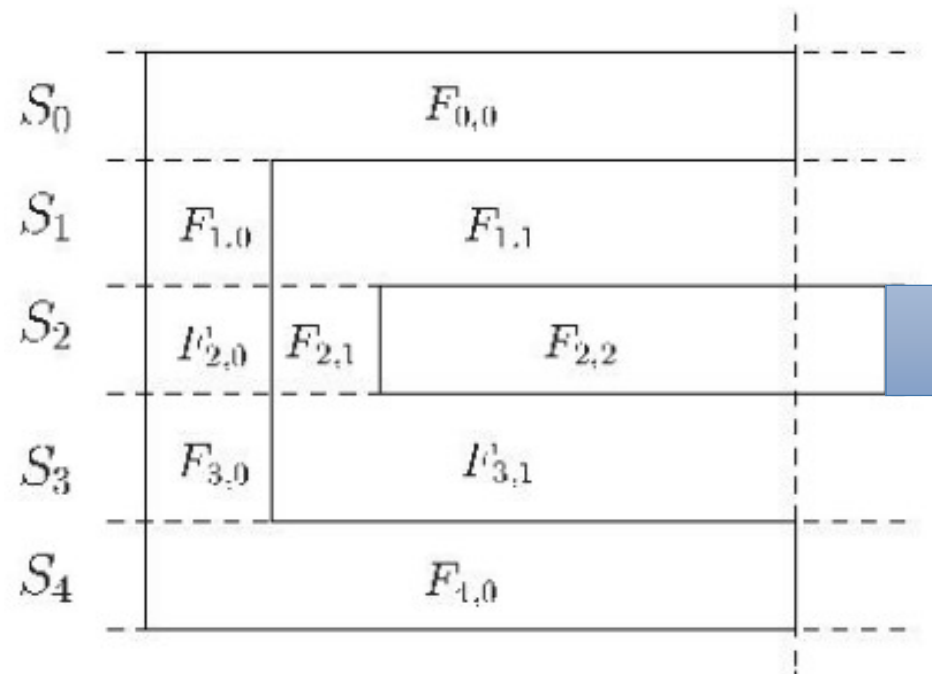
Represent layout as layout matrix

- 将layout划分成网格，如果一个长方形与网格重叠，则该网格=1，反之网格=0，这样得到layout matrix；
- 这样的表示能够寻找到所有匹配结果，但是不高效，使得匹配算法计算复杂且昂贵；
- Cutting-slice 将被表示区域划分成一组水平或垂直的slices集合 $\{S_0, S_1, \dots, S_{n-1}\}$ ，且相邻的slices不相等；
- 每个slice S_i 被划分成若干fragments $\{F_{i,0}, F_{i,1}, \dots, F_{i,n-1}\}$ ，且相邻的fragments也不相等；
- 表示的网格大小可以进行调整，在层次化的匹配中，先使用大网格进行粗匹配，再使用小网格进行细匹配。



Cutting-slice Representation

- 将要搜索的block和要被匹配的pattern都用 cutting-slice表示成1维string；
- Slice的切割方向尽量与延展方向相同，使拓扑与延展前相同，前后能够使用同一个string；
- 对每个slice根据fragment的值进行编码，将所有slices拼接起来组成最后的编码。每个slice的起始位置以 '1' 开始。



Cutting-slices of range pattern

S_0 : '11' = '3'

S_1 : '110' = '6'

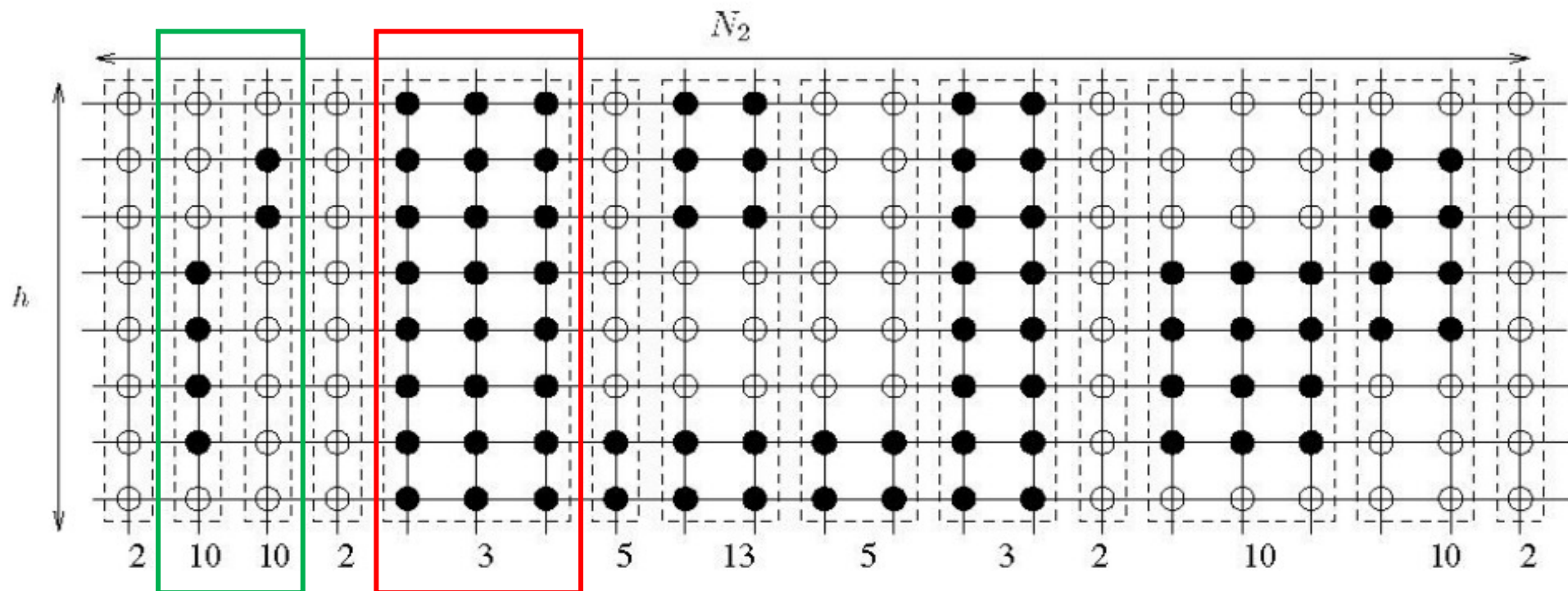
S_2 : '1101' = '13'

S_3 : '110' = '6'

S_4 : '11' = '3'

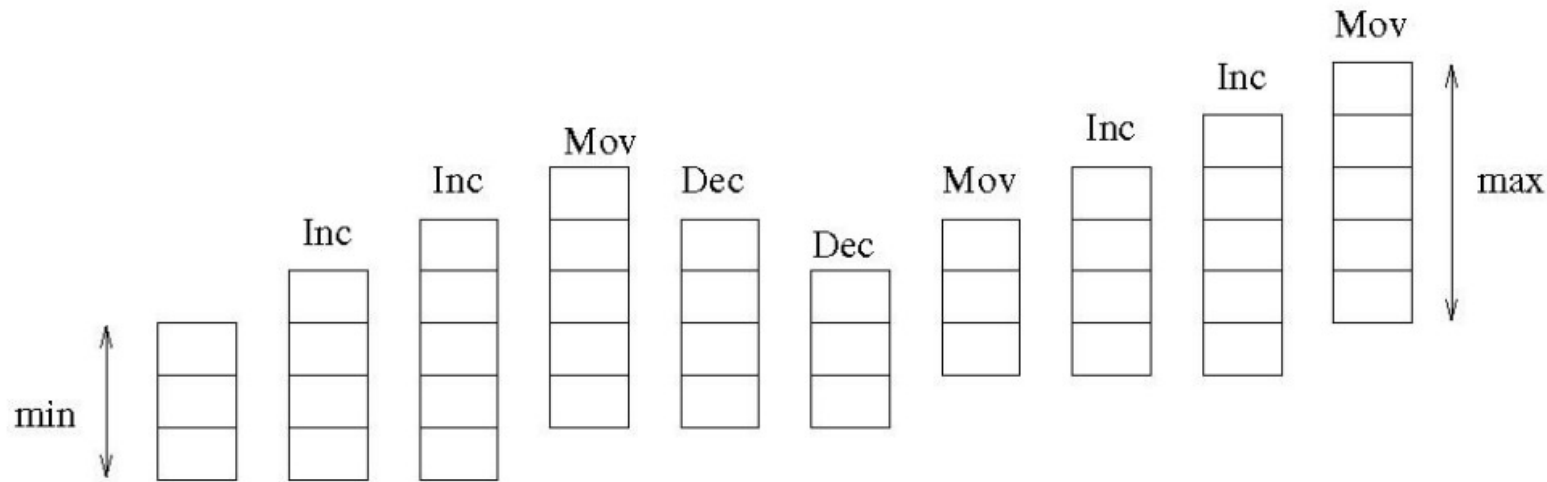
Pattern: {3, 6, 13, 6, 3}

Cutting-slice Representation



- 相同的slices会进行合并，如红色框中所示；
- 绿色框中的两条slices虽然编码相同，但是grids的表示并不相同，所以不能合并；
- 通过这样的表示，上图block就可以用 $\{2, 10, 10, 2, 3, 5, 13, 5, 3, 2, 10, 10, 2\}$ 一维向量进行表示；
- 最后，使用KMP算法 [2] 对得到的block和pattern的一维string进行快速匹配。

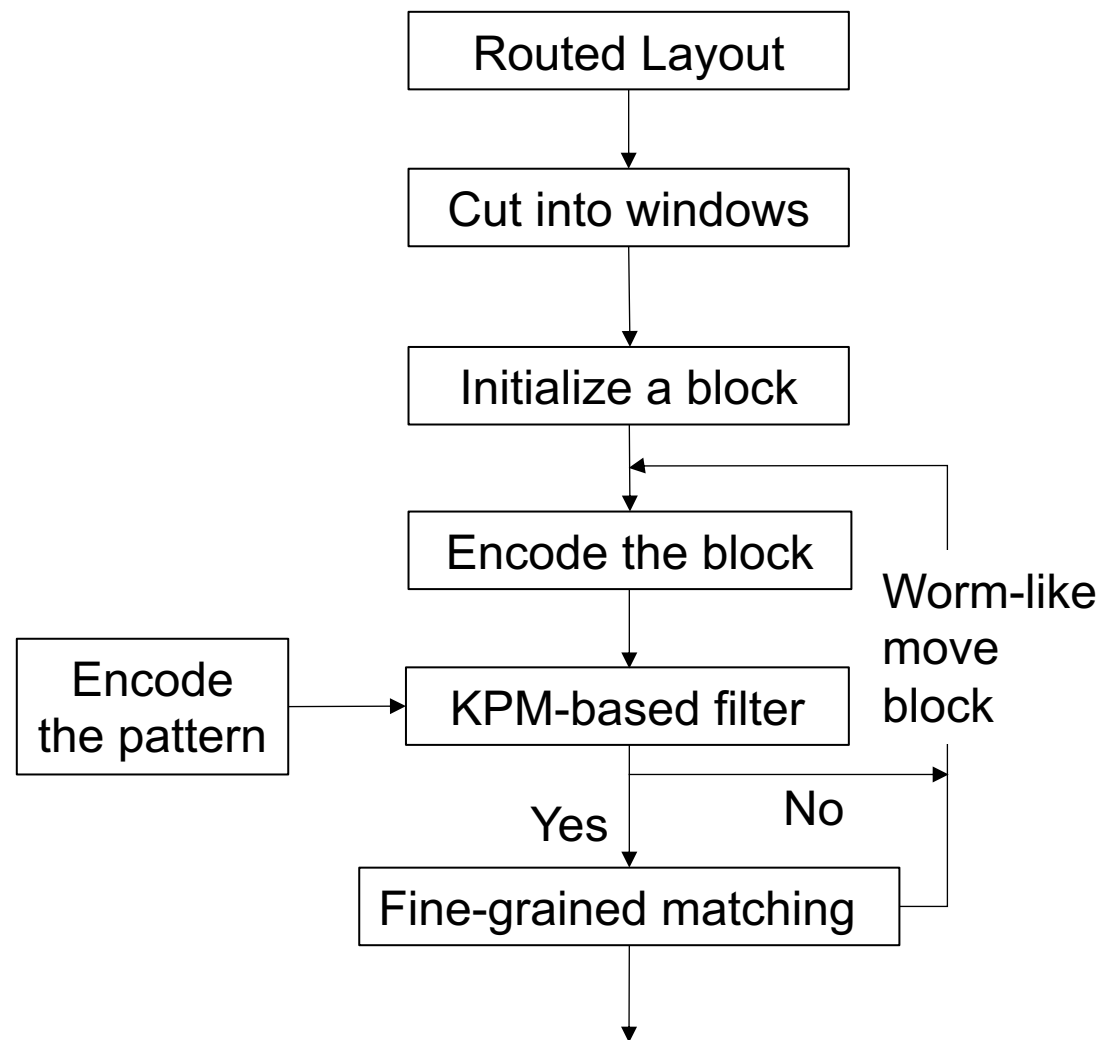
Cut layout matrix into blocks



- 在layout上移动block进行上述匹配，第一个block从layout的最后一行开始，然后在layout上进行worm-like移动。
- 如使用垂直的cutting-slice表示，block的列数等于layout的宽度。Block的高度需在range pattern的可变范围内。

赛题解析

- 用cutting-slice表示range pattern和layout；
- 将layout划分为若干窗口，在每个窗口里worm-like移动block搜索整个窗口；
- 在每个block中进行层次化匹配。粗网格匹配找到可能matching的位置，再使用细网格进行精细匹配。
- 对于旋转和镜像变化，产生新的range pattern重新和layout进行匹配。



优化方向

- 减少存储量
- 并行性
- 实现效率
- 将layout表示成图，使用DRC检测的方法 [3]
- 使用深度学习目标检测的方法

参考文献

- [1] Yao H, Sinha S, Chiang C, et al. Efficient process-hotspot detection using range pattern matching[C]//Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design. 2006: 625-632.
- [2] Knuth D E, Morris, Jr J H, Pratt V R. Fast pattern matching in strings[J]. SIAM journal on computing, 1977, 6(2): 323-350.
- [3] Yu Y T, Jiang H R, Zhang Y, et al. DRC-based hotspot detection considering edge tolerance and incomplete specification: U.S. Patent 9,594,867[P]. 2017-3-14.