

# 2023（第五届）集成电路EDA设计精英挑战赛

## 赛题指南

版本	时间	修订内容
V1.0	2023/8/18	初版发布
V1.1	2023/10/31	更新第六章第3项:多线程并行计算评分标准

### 一、 赛题名称

超大规模版图图形匹配算法

### 二、 命题企业

杭州广立微电子股份有限公司

### 三、 赛题 Chair

杨帆（复旦大学）

### 四、 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，其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，因此，一种专用图形匹配工具就被提出来。

目前，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<sup>[1]</sup>，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<sup>[2]</sup>，重复单元图形加速 OPC 修正<sup>[3]</sup>等多个领域。

## 五、赛题描述

图形匹配的最基础应用即为根据模板库中的模板图形，在版图文件搜索并定位模板图形在其中的位置，本赛题的目标即为实现此功能。

### 1. 输入文件描述：

本赛题将提供 1 个版图文件和 3 个模板图形文件。

1) 模板图形文件：由若干个曼哈顿多边形（所有的边均平行或垂直于平面坐标轴，以下简称为多边形）和一个关键区域标记（矩形框）组成。图 1a 为一个模板图形的示例，其由两个多边形（蓝色图形）和一个关键区域标记（橙色框）组成。本赛题用文本文件来描述模板图形，其格式如图 1b 所示。其包含了两部分，第一部分描述了模板中包含的多边形的信息。以 pattern 开头（第 1 行）。紧随其后的两行分别按逆时针方向描述了一个多边形各个顶点的坐标。其中，第 2 行描述了#1 多边形（F 形多边形），该多边形包含 10 个顶点，(0, 0) 为起始点坐标，(500, 0) 为第二个顶点坐标，

依次类推。第 3 行描述了#2 多边形（小矩形），该多边形包含 4 个顶点。第 4 行开始进入第二部分，描述了关键区域标记的信息。以 marker 开头。紧随其后的第 5 行按逆时针方向描述了关键区域标记框 4 个顶点的坐标。请注意以下几点：

- a) 每个模板有且仅有一个关键区域标记；
- b) 关键区域标记为矩形；
- c) 模板内的所有多边形的边均落在关键区域标记内或与关键区域标记的边重合；

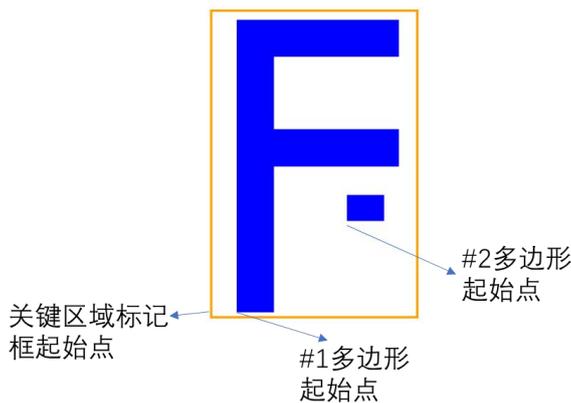


图 1a

```

1 pattern:
2 (0,0), (500,0), (500,2000), (2200,2000), (2200,2500), (500,2500), (500,3500), (2200,3500), (2200,4000), (0,4000)
3 (1500,1250), (2000,1250), (2000,1600), (1500,1600)
4 marker:
5 (-350,-75), (2450,-75), (2450,4130), (-350,4130)

```

图 1b

如果模板图形文件中出现被 [ ] 包围的多边形顶点，如图 2b 第 3 行，则该行多边形与上一行共同描述了一个可变形状的多边形。如图 2a，实心深蓝色 F 形为图 2b 第 2 行描述的多边形，红色斜线 F 形为图 2b 第 3 行描述的多边形。则该模板图形要求版图中的此类多边形只要边 e 落在区域 a 的范

围内（包含范围边界）均算作与模板匹配。

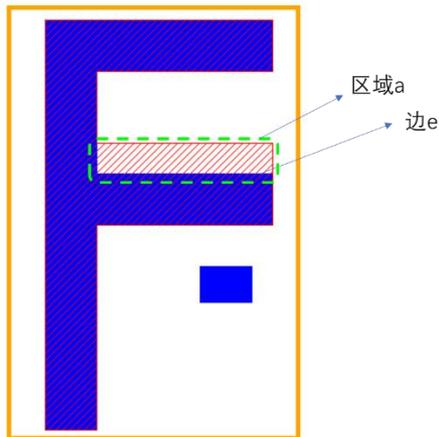


图 2a

```

1 pattern:
2 (0,0), (500,0), (500,2000), (2200,2000), (2200,2500), (500,2500), (500,3500), (2200,3500), (2200,4000), (0,4000)
3 [(0,0), (500,0), (500,2000), (2200,2000), (2200,2800), (500,2800), (500,3500), (2200,3500), (2200,4000), (0,4000)]
4 (1500,1250), (2000,1250), (2000,1600), (1500,1600)
5 marker:
6 (-350,-75), (2450,-75), (2450,4130), (-350,4130)

```

图 2b

2) 版图文件：由若干个曼哈顿多边形（以下简称为多边形）组成，且数量远多于模板图形文件中的数量。图 3 为一个小型版图的示例，其由 5 个多边形组成。本赛题为了减轻参赛者解析版图文件的负担，采用文本文件，而非标准的 GDSII 或 OASIS 文件来描述版图，其格式如图 4 所示。每一行按逆时针方向描述了一个多边形各个顶点的坐标。1~5 行分别对应了图 3 中#1~#5 多边形。

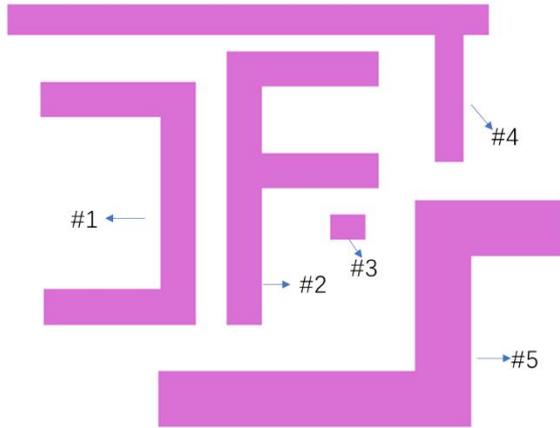


图 3

1	(34236,19338), (35974,19338), (35974,16804), (34272,16804), (34272,16295), (36474,16295), (36474,19838), (34236,19838)
2	(36947,16295), (37447,16295), (37447,18295), (39147,18295), (39147,18795), (37447,18795), (37447,19795), (39147,19795), (39147,20295), (36947,20295)
3	(38447,17545), (38947,17545), (38947,17895), (38447,17895)
4	(33747,20553), (39974,20553), (39974,18685), (40383,18685), (40383,20553), (40753,20553), (40753,20981), (33747,20981)
5	(35946,14801), (40491,14801), (40491,17312), (41842,17312), (41842,18112), (39691,18112), (39691,15601), (35946,15601)

图 4

## 2. 图形匹配要求:

- 依次以 3 个模板文件中的 pattern 为模板，在版图文件中搜索图形，输出**所有**完全匹配的结果，匹配结果以 3 个模板文件中关键区域标记 (marker) 在版图文件中 pattern 匹配到的相对坐标形式输出 (marker 与 pattern 的相对位置和模板文件中保持一致, 顶点按逆时针顺序排列)。例如，图 5 展现了用图 1 的模板在图 3 的版图匹配的结果, 输出结果为图 5 中关键区域标记 (橙色框) 4 个顶点 (按逆时针方向排序) 的坐标, 此时结果应为:“(36597, 16220), (39397, 16220), (39397, 20425), (36597, 20425)”。

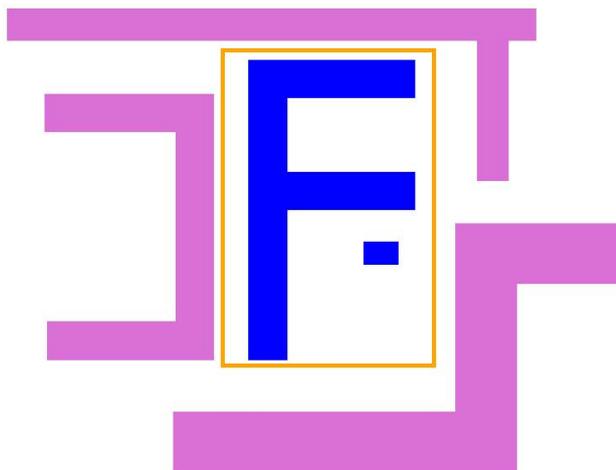


图 5

2) 依次以 3 个模板文件中的 pattern 为模板，在版图文件中搜索图形，除了完全匹配之外，加入以下版图发生旋转和镜像情况下的匹配结果，如图 6：

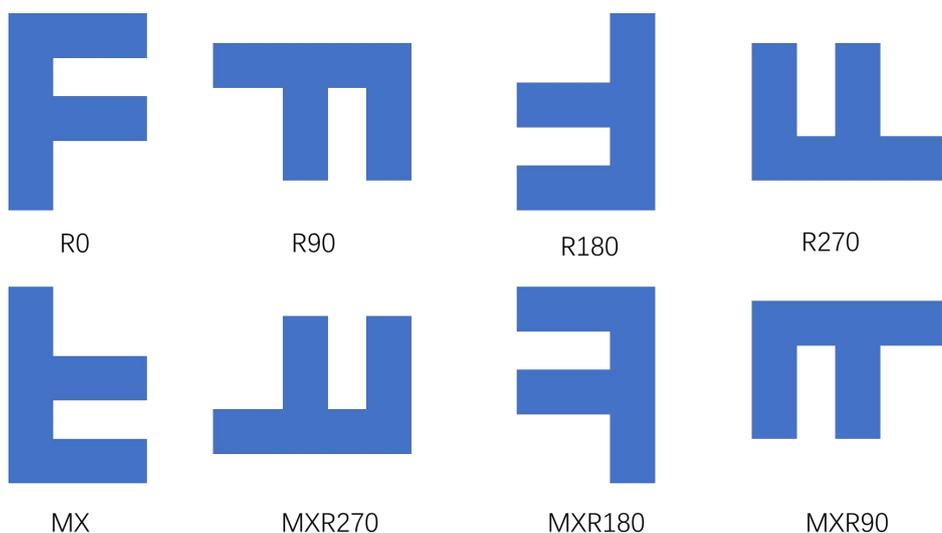


图 6

则如下图 7 的版图中也应能匹配到图 1 的模板，并输出匹配后的关键区域标记（橙色框）4 个顶点的坐标，此时结果应为：“（39127,16854），（34922,16854），（34922,19654），（39127,19654）”。

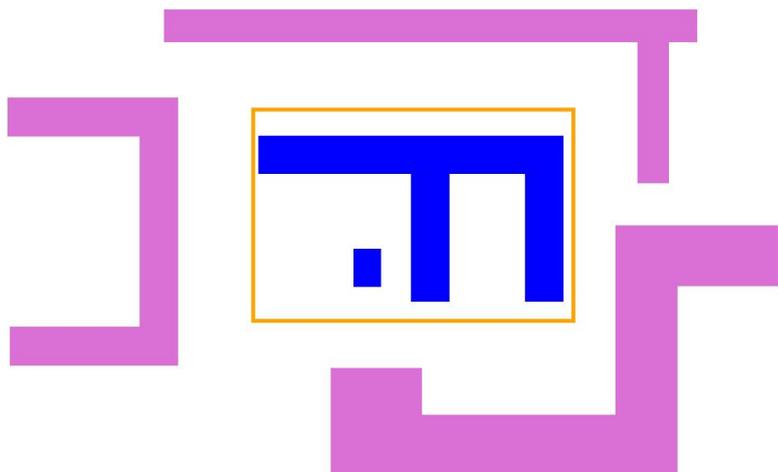


图 7

3) 注意，版图匹配时仅关注模板关键区域标记内版图的形状，不关注其他区域的版图环境。如在模板的多边形中存在与关键区域标记的边重合的边，则下图 8 中的两种版图情形均视为匹配：

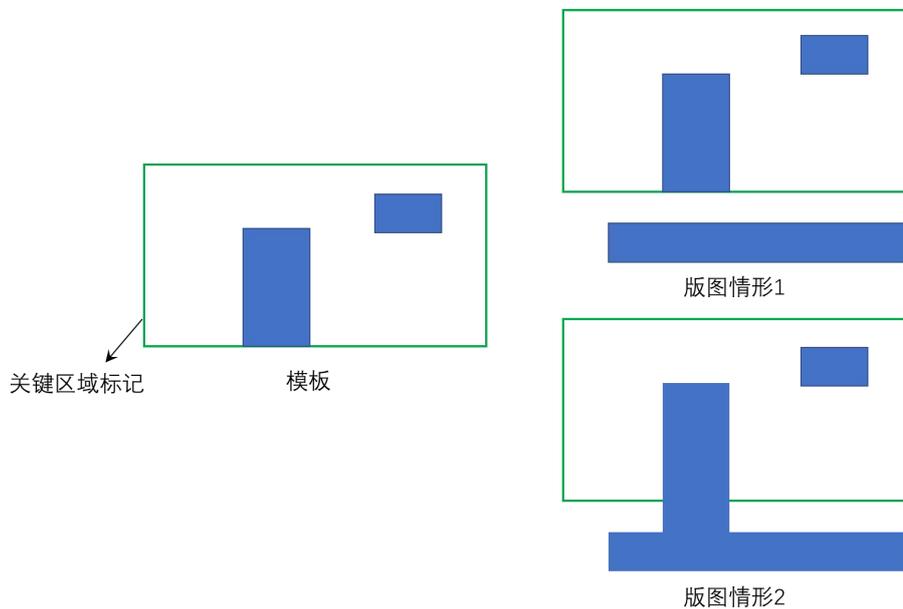


图 8

4) 可以使用多线程并行计算技术加速图形匹配。

### 3. 输出文件格式：

输出文件格式为 linux 系统的文本文件格式。内容为当前模板图形文件在当前版图文件匹配到的所有结果的合集，每一行保存一个结果，结果的记录方式参见图形匹配要求章节。

### 4. 运行语法：

```
pm -layout ./layout.txt -lib ./lib.txt  
-with_rot_mir Y/N [-thread n] -output ./res.txt
```

pm: 图形匹配 (Pattern match) 可执行程序，由参赛者在大赛服务器上编译而来。

参数设置：

-layout: 必选参数，指定版图文件的路径，该文件由出题方提供。

-lib: 必选参数，指定模板图形文件的路径，该文件由出题方提供。

-with\_rot\_mir: 必选参数，指定当前图形匹配时是否考虑包含旋转和镜像的 8 情况 (参见图 6)。指定 N 时表示只考虑完全匹配的情况，指定 Y 时表示考虑图 6 中所有的 8 种情况。

-thread: 可选参数，指定 n 个线程的并行计算。

-output: 必须参数，指定输出的结果文件的路径，由程序运行产生。

## 5. 提交内容:

所有参赛队伍完成赛题后须提交以下内容:

- 1) 编译好的可执行程序, 运行库 (如有)
- 2) 所有程序源代码
- 3) 开发中所有调用的第三方库的信息: 库名及其官网

## 六、 评分标准

本赛题的得分点分为 3 部分: 分别为正确性得分, 性能得分和多线程并行计算得分。各部分评分标准如下:

1. 正确性: 满分 60 分。本赛题共 3 个匹配模板图形库, 每个图形库又分为完全匹配和考虑旋转和镜像的匹配, 共 6 个匹配结果。每个匹配结果完全正确, 无错误匹配或漏匹配, 则得 10 分。每有 1 个错误匹配或漏匹配, 扣 1 分, 扣完为止。将 6 个匹配结果的得分相加, 所得分数即为正确性得分。
2. 性能: 满分 30 分。仅在正确性得分中得了 10 分的匹配结果才可能得性能分。性能分按该项匹配的**单进程单线程**运行时间排名来计算分值。对所有获得该项匹配正确性得 10 分的参赛选手的运行时间按从短到长的顺序进行排序, 第 1 名得 5 分, 此后名次每降低一名, 得分扣 5/(参与排名人数-1)。将 6 个匹配结果的得分相加, 所得分数即为性能得分。
3. 多线程并行计算: 满分 10 分。本赛题考察的是 8 个线程

并行时的并行效率，未使用多线程并行计算的参赛选手本项不得分。并行效率的计算方法为：并行效率=单线程运行时间/(多线程运行时间\*线程数)。本项目得分的前提是：1) 正确性得满分60分，2) 单线程与多线程运行所得结果文件完全一致，3) 8个线程并行时的并行效率 $>0.6$ 且 $\leq 1$ 。对并行效率按从高到低的顺利排序，按名次，第1名得10分，此后名次每降低一名，得分扣 $10/(\text{参与排名人数}-1)$ 。

## 七、参考资料

[1] Izumi Nitta, Yuzi Kanazawa, Tsutomu Ishida, and Koji Banno "A fuzzy pattern matching method based on graph kernel for lithography hotspot detection", Proc. SPIE 10148, Design-Process-Technology Co-optimization for Manufacturability XI, 101480U (28 March 2017); <https://doi.org/10.1117/12.2257654>

[2] Uwe Paul Schroeder, Janam Bakshi, Ahmed Mounir Elsemary, and Fadi Batarseh "Dynamic pattern matching flow to enable low escape rate weak point detection", Proc. SPIE 11328, Design-Process-Technology Co-optimization for Manufacturability XIV, 113280D (23 March 2020); <https://doi.org/10.1117/12.2551739>

[3] Piyush Verma, Fadi Batarseh, Shikha Somani, Jingyu Wang, Sarah McGowan, and Sriram Madhavan "Pattern-based pre-OPC operation to improve model-based OPC runtime", Proc. SPIE 9235, Photomask Technology 2014, 923506 (8 October 2014); <https://doi.org/10.1117/12.2068998>