2024中国研究生创“芯”·EDA精英挑战赛

赛题指南

一、 赛题名称

多层图形匹配与设计单元级版图验证算法

二、 命题企业

杭州广立微电子股份有限公司

三、 赛题主席

姚海龙（北京科技大学）

四、赛题背景

图形匹配（Pattern matching）技术在芯片的可制造性设计（DFM）、版图物理验证、良率提升等领域已得到了广泛的应用[1][2][3]。在版图物理验证领域，图形匹配技术可以弥补传统 DRC 工具对复杂结构版图验证能力的不足，特别在验证多层版图图形结构的正确性上，图形匹配技术优势明显。

SRAM 区域是逻辑芯片中器件密度最高的区域之一，也是设计容错空间最小的区域之一，传统的 DRC 版图验证方法在对 SRAM 区域进行检查时，不仅计算资源消耗大，而且很难有效覆盖所有设计失误类型。因同一芯片内 SRAM 单元（SRAM Bitcell）的种类较少，采用基于图形匹配技术的物理验证方法可以大幅减小计算资源开销[4][5]，同时显著提高设计失误类型覆盖率。目前，这种基于图形匹配技术的

SRAM 物理验证方法已在工业界得到了广泛应用。

五、赛题描述

本赛题将模拟工业界基于图形匹配技术的 SRAM 物理验证应用场景，完成对指定版图的设计单元匹配，并找出存在错误的设计单元。输入为待查版图、设计单元模板（类似 SRAM bitcell），输出为错误设计单元坐标，错误设计单元中的错误层，错误层图形与该层正确图形的差异。

1. 输入文件描述：

本赛题将提供 1 个版图文件和 1 个**包含多个模板图形**的模板图形文件。

1） 模板图形文件：模板图形由若干层的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 1 为一个模板图形的示例，它由不同层的多个多边形（红色、绿色、蓝色、黄色、橙色图形）和一个关键区域标记（橙色框）组成。

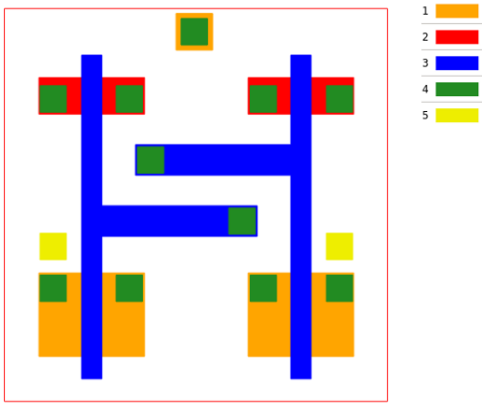

图 1 模板图形示例

本赛题用文本文件来描述模板图形，其格式如图 2 所示。每个模板图形以模板标识（pattern1，pattern2，…）作为开头，到下一个模板标识或者文件结尾，模板图形内容包含两部分，第一部分是不同层的多边形信息。每层的多边形信息以层标识开头（layer1, layer2, …），到下一个层标识或者关键区域标识。这之间的每一行按逆时针方向给出了一个多边形各个顶点的坐标。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) 每个模板图形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；

```
pattern1:
layer1:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
layer2:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
marker:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
pattern2:
layer1:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
layer2:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
layer3:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
marker:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
```

图 2 模板图形的文本描述

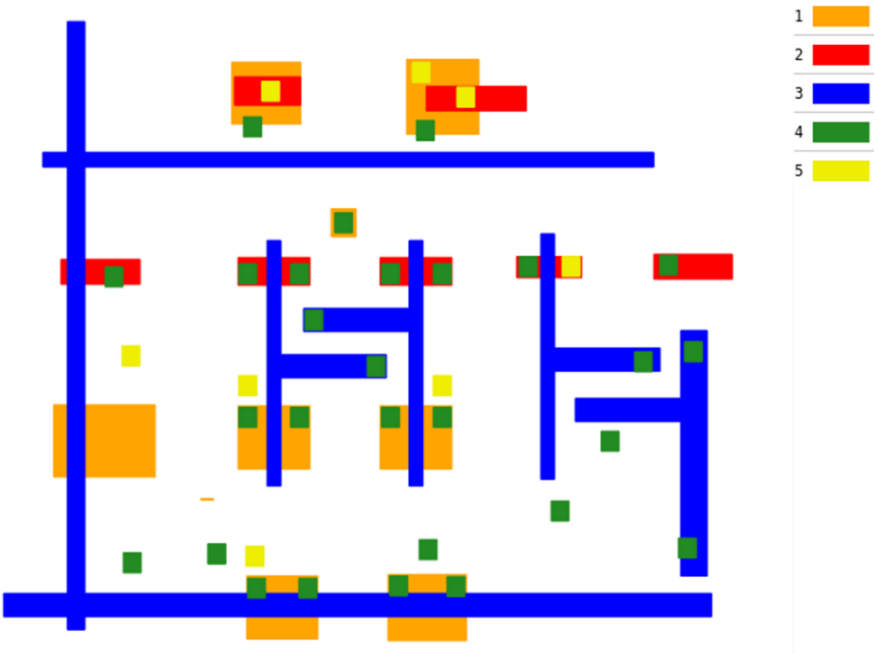

图 3 小型版图示例

2）版图文件：由若干层的若干个曼哈顿多边形（以下简称为多边形）组成，且其多边形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。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。版图文件中包含多层的多边形，每层的多边形以层标识开头，到下一个层标识或者文件结尾，层标识之间的每一行按逆时针方向描述了一个多边形各个顶点的坐标。

.\layout.txt

```
layer1:
(-70,1135),(450,1135),(450,1545),(-70,1545)
(605,2640),(785,2640),(785,2820),(605,2820)
(960,1135),(1480,1135),(1480,1545),(960,1545)
(1016,23),(1586,23),(1586,452),(1016,452)
(1858,3277),(2357,3277),(2357,3585),(1858,3585)
(3125,3211),(3648,3211),(3648,3696),(3125,3696)
(3508,1461),(3748,1461),(3748,1928),(3508,1928)
(3263,-22),(3748,-22),(3748,69),(3263,69)
layer2:
(-70,2325),(450,2325),(450,2505),(-70,2505)
(960,2325),(1480,2325),(1480,2505),(960,2505)
(1877,3399),(2357,3399),(2357,3583)(1877,3583)
(1949,2374),(2420,2374),(2420,2511),(1949,2511)
(3267,3362),(3748,3362),(3748,3522),(3267,3522)
(2943,2365),(3513,2365),(3513,2525),(2943,2525)
(3644,1149),(3748,1149),(3748,1309),(3644,1309)
layer3:
(140,1025),(240,1025),(240,1725),(1005,1725),(1005,1875),(240,1875),(240,2615),(140,2615)
(405,2025),(1170,2025),(1170,1025),(1270,1025),(1270,2615),(1170,2615),(1170,2175),(405,2175)
(2124,1068),(2224,1068),(2224,1768),(2989,1768),(2989,1918),(2224,1918),(2224,2658),(2124,2658)
(1548,173),(1843,173),(1843,-22),(2046,-22),(2046,173),(3748,173),(3748,267),(1548,267)
(2374,1442),(3139,1442),(3139, 442),(3331, 442),(3331,2032),(3139,2032),(3139,1592),(2374,1592)
(3587,2166),(3748,2166),(3748,2508),(3587,2508)
layer4:
(3196,3170),(3326,3170),(3326,3300),(3196,3300)
(3165,1830),(3295,1830),(3295,1960),(3165,1960)
(3122,558),(3252,558),(3252,688),(3122,688)
(2985,2391),(3115,2391),(3115,2521),(2985,2521)
(2802,1765),(2932,1765),(2932,1895),(2802,1895)
(2561,1250),(2691,1250),(2691,1380),(2561,1380)
(2199,798),(2329,798),(2329,928),(2199,928)
(2022,2847),(3748,2847),(3748,3069),(2022,3069)
(1967,2382),(2097,2382),(2097,2512),(1967,2512)
(1444,308),(1574,308),(1574,438),(1444,438)
(1345,2335),(1475,2335),(1475,2465),(1345,2465)
(1345,1405),(1475,1405),(1475,1535),(1345,1535)
(1242,548),(1372,548),(1372,678),(1242,678)
(1029,313),(1159,313),(1159,443),(1029,443)
(970,2335),(1100,2335),(1100,2465),(970,2465)
(970,1405),(1100,1405),(1100,1535),(970,1535)
(865,1735),(995,1735),(995,1865),(865,1865)
(667,1183),(797,1183),(797,1313),(667,1313)
(630,2665),(760,2665),(760,2795),(630,2795)
(415,2035),(545,2035),(545,2165),(415,2165)
(310,2335),(440,2335),(440,2465),(310,2465)
(310,1405),(440,1405),(440,1535),(310,1535)
(-65,2335),(65,2335),(65,2465),(-65,2465)
(-65,1405),(65,1405),(65,1535),(-65,1535)
(-289,520),(-159,520),(-159,650),(-289,650)
(-902,463),(-772,463),(-772,593),(-902,593)
(-1049,2214),(-904,2214),(-904,2437),(-1049,2437)
layer5:
(2279,2382),(2409,2382),(2409,2512),(2279,2512)
(1515,3479),(1645,3479),(1645,3609),(1515,3609)
(1345,1610),(1475,1610),(1475,1740),(1345,1740)
(1191,3639),(1321,3639),(1321,3769),(1191,3769)
(101,3517),(231,3517),(231,3647),(101,3647)
(-13,505),(117,505),(117,635),(-13,635)
(-911,1802),(-781,1802),(-781,1932),(-911,1932)
```

图 4 版图文件示例

2. 图形匹配要求：

本赛题需要依次以模板文件中的多个 pattern 为模板，在版图文件中进行多层图形搜索，输出所有部分匹配的结果。部分匹配的定义为：假设 pattern 中共有 n 层（n>3），其中至少有三层的图形匹配为完全匹配，其中某一层完全匹配指的是在版图上使用与 pattern 相同的 marker 形状框选出的区域

内，图形集合与 pattern 中该层的图形集合完全一致。例如，图 6 中的黄色斜线框（marker）内第 1-3 层中的多边形集合与图 1 中黄色框内的图形集合完全相同，说明第 1-3 层都是完全匹配的，第 4 层和第 5 层都存在一些不同的多边形。因此该区域就是部分匹配的结果之一。

部分匹配的结果输出包括部分匹配的关键区域标记(marker)和部分匹配区域内的多边形集合和对应 pattern 中的多边形集合之间的差异。该差异结果为两组多边形进行异或运算的结果。图形异或运算的定义如下：多边形 A 与多边形 B 的异或（XOR）操作结果是一个新的多边形集，它包括以下区域：1. 仅在多边形 A 内但不在多边形 B 内的部分。2. 仅在多边形 B 内但不在多边形 A 内的部分。即异或操作将两个多边形相交的区域去除，只保留那些只属于其中一个多边形的区域（如图 5 右下角所示）。

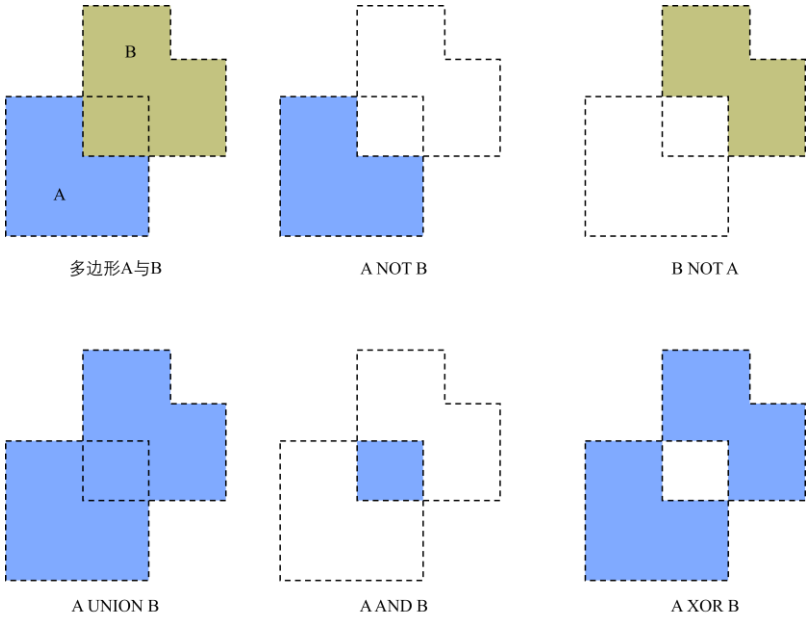

图 5 XOR 操作示意图

图 6 是利用图 1 所示的模板（pattern）在图 3 所示的版图文件中进行匹配得到的部分匹配结果。该结果的输出如图 7 所示。其中“pattern1”是图 1 所示的 pattern 的名称，marker 的坐标为图 6 中橙色斜线框的坐标，以逆时针方向表示。layer4 和 layer5 层号后面紧跟的是第 4 层多出的矩形和第 5 层缺少的矩形的坐标，同样以逆时针方向表示。具体的输出格式详见后续章节说明。

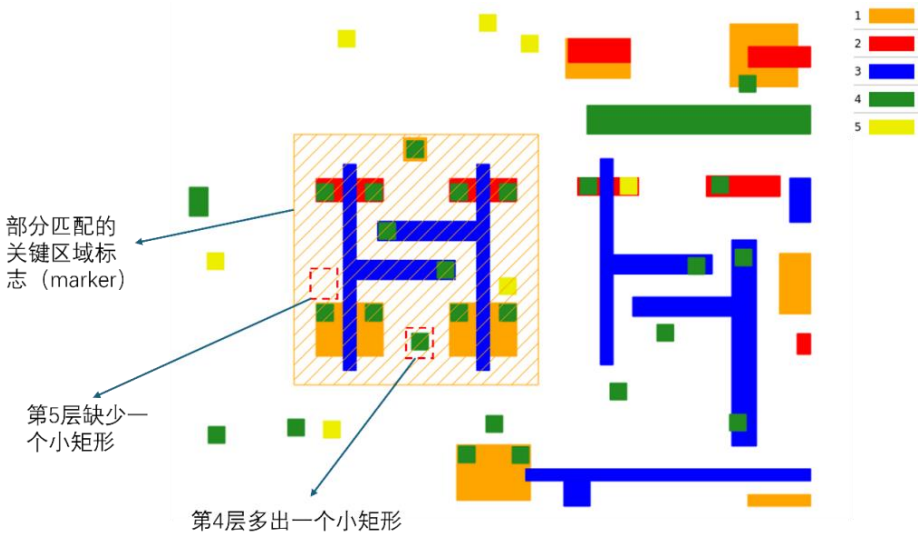

图 6 匹配结果示意图

res.txt

```
pattern1:
marker:
(-240,914),(1646,914),(1646,2844),(-240,2844)
layer4:
(667, 1183),(797,1313),(797,1313),(667,1313)
layer5:
(-65,1610),(65,1610),(65,1740),(-65,1740)
```

图 7 输出结果示例

除了与原始模板（pattern）完全一致的匹配结果之外，本赛题还要求匹配旋转和镜像之后的匹配结果。所有的旋转镜像方向如图 8 所示。图 9 则给出了图 1 所示 pattern 在选转镜像条件下的部分匹配结果示意图。

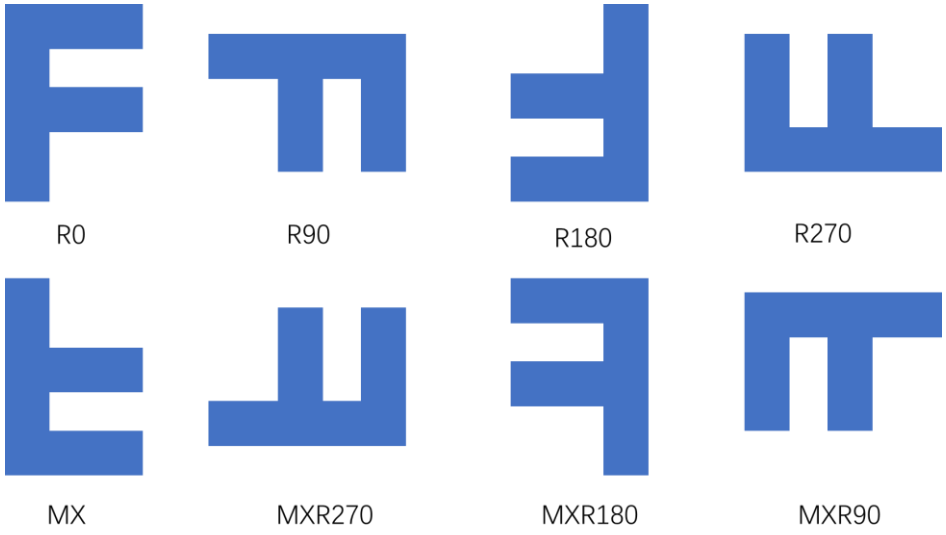

图 8 旋转镜像示意图

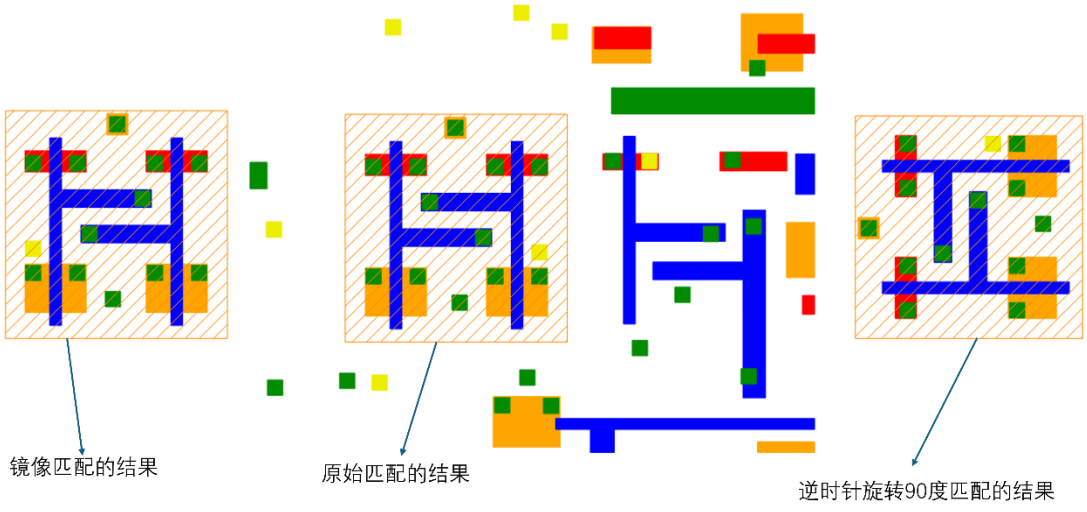

图 9 旋转镜像后的部分匹配结果示意图

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 10 中两种版图情形均视为匹配：

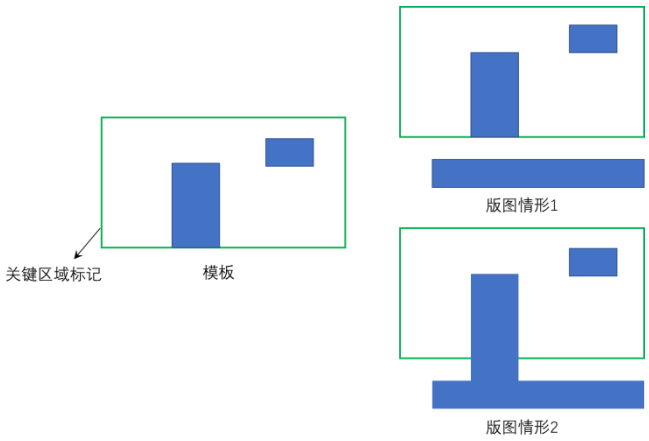

图 10 两种版图匹配情形示例

3. 输出文件格式：

输出文件格式为 linux 系统的文本文件格式。内容为部分匹配的版图区域的相关信息，包括关键区域标记（marker）的坐标、不匹配的图层、不匹配图层图形与模板（pattern）中对应层正确图形的异或结果。

错误图形的信息按照所属模板图形不同分别输出，输出一个模板图形下的错误图形前由模板标识开头（pattern1, pattern2, …），每个错误图形的信息通过多行输出，第一行是错误图形的标记框标识 marker，下一行是该标记框坐标，再下一行是第一个错误层的标识，接下来的行是该错误层与正确图形的差异，用一系列多边形输出，每行一个多边形，这一层的所有差异输出完之后输出下一个错误层（如果有），以此类推，直到输出所有错误层和该层与正确图形之间的差异，

再接着输出下一个错误图形，以此类推，直到输出完所有错误图形，然后输出下一个模板图形下的错误图形，以此类推，直到将所有模板对应的错误图形全部输出。输出格式如图 11 所示。

```
pattern1:
marker:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
layer1:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
layer3:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
marker:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
layer1:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
layer2:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
pattern2:
marker:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
layer1:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
layer3:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
marker:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
layer1:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
layer2:
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
(xxx,xxx),(xxx,xxx),(xxx,xxx),(xxx,xxx)
```

图 11 匹配结果输出格式

4. 运行语法：

pm -layout ./layout.txt -lib ./lib.txt [-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。

-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）开发中所有调用的第三方库的信息：库名及其官网

3）鼓励提交源码

六、评分标准

本赛题的得分点分为两部分：分别为正确性得分，性能得分。各部分评分标准如下：

1. 正确性：

满分 60 分，单项分值为 60 分除以应匹配到的匹配区域结果的个数。例如，应匹配到的匹配区域结果有 12 个，则单项的分值为 60/12=5 分。匹配结果无漏报、无误报则得满分

60 分。分数扣完为止，即最低得分为 0 分。

漏报定义为输出的 marker 坐标集合未覆盖全部应匹配到的匹配区域位置。每遗漏一处，扣除一份单项分值。例如，单项分值为 5 分，提交作品的结果共有 2 处匹配区域位置遗漏，则正确性得分扣除 5*2=10 分。

误报分为两种情况：（1）输出的匹配区域的 marker 坐标不是正确的匹配区域位置，此种情况每误报一处，扣除一份单项分值；（2）输出的匹配区域的 marker 坐标正确，但输出的与模板比较的差异结果信息错误（包括差异层输出错误和异或结果输出错误），此种情况每误报一处，扣除一半的单项分值。例如，单项分值为 5 分，第一种情况的误报 1 处，第二种情况的误报 3 处，则正确性得分扣除 5*1+5*0.5*3=12.5 分。

2. 性能：

满分 40 分。分为单线程和多线程两部分，单线程 20 分，多线程 20 分，在正确性得分中得了满分才可得性能分。

单线程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。对所有获得正确性满分的参赛选手的运行时间按从小到大的顺序进行排序，第 1 名得 20 分，此后名次每降低一名，得分扣 20/(参与排名人数-1)。**若在未设定-thread 选项的情况下程序执行时使用了多进程或多线程的方式，则该项得分计为 0 分。**

多线程性能分按该项匹配的**单进程 8 线程（保证运行环境 CPU 核心数不少于 8）**运行时间排名来计算分值，**若未实现多线程，使用单线程的运行时间来计算分值**。对所有获得正确性满分的参赛选手的运行时间按从小到大的顺序进行排序，第 1 名得 20 分，此后名次每降低一名，得分扣 20/(参与排名人数-1)。**若程序执行时使用的最大线程数超过了-thread 选项设定的线程数，则该项得分计为 0 分。**

3. 其他说明：

若参赛作品在大赛服务器的运行环境中运行测评案例（Hidden case）单进程单线程的运行时间超过 8 小时，则既无法获得正确性得分，也无法获得性能得分，即各项得分均为 0 分。

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

[4] H. Yao, S. Sinha, C. Chiang, X. Hong and Y. Cai, "Efficient Process-Hotspot Detection Using Range Pattern Matching," 2006 IEEE/ACM International Conference on Computer Aided Design, San Jose, CA, USA, 2006, pp. 625-632, doi:

10.1109/ICCAD.2006.320026.
[5] J. W. Park, R. Todd and X. Song, "Geometric Pattern Match Using Edge Driven Dissected Rectangles and Vector Space," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 35, no. 12, pp. 2046-2055, 2016, doi: 10.1109/TCAD.2016.2535908.

*本赛题指南未尽问题，见赛题 Q&A 文件