2025 中国研究生创“芯”大赛·EDA 精英挑战赛

一、赛题名称

超大规模版图模板分类算法

二、命题单位

深圳市海思半导体有限公司

三、赛题背景

设计制造迭代过程中，一些可能造成芯片缺陷的图形模板(Pattern)被记录下来用于分析芯片制造性能与良率。随着超大规模集成电路(VLSI)的工艺演进，Pattern 数量通常达到百万至数亿级别，而这些 Pattern 中往往存在大量严格相同或者高度相似的 Pattern。通过模板分类(Pattern Cluster)技术可以大幅度降低 Pattern 的数量级，同时可以优先处理重复率高的 Pattern。

近年来，模板分类技术被广泛应用于重复图形的 OPC 修正，版图场景覆盖分析，设计规则生成等多个领域。模板分类在 EDA 软件中具有提升复杂版图分析效率、加速设计和验证过程以及挖掘潜在设计创新点等多方面价值。高效高精度的模板分类算法可以作为 EDA 软件的核心亮点，提升软件在行业中的竞争力。

四、赛题描述

模板分类的目标是提取版图中的潜在热点图形，通过特征提取或者规则比较等技术将这些热点进行归类。本赛题重点关注模板分类算法实现的精度、性能及峰值内存等指标。

1. 输入文件描述：

1） 版图文件(gds 或 oas 格式)，文件中包含两层，如下图所示：

① 图形设计层(Design Layer)，层编号固定为(1,0)，设计层中包含大量曼哈顿图形，即图形边必然与 x 轴或 y 轴平行；

② 潜在热点标记层(Marker Layer)，层编号固定为(2,0)，标记层由大量矩形组成，每个矩形代表 Pattern 中心的选取范围。

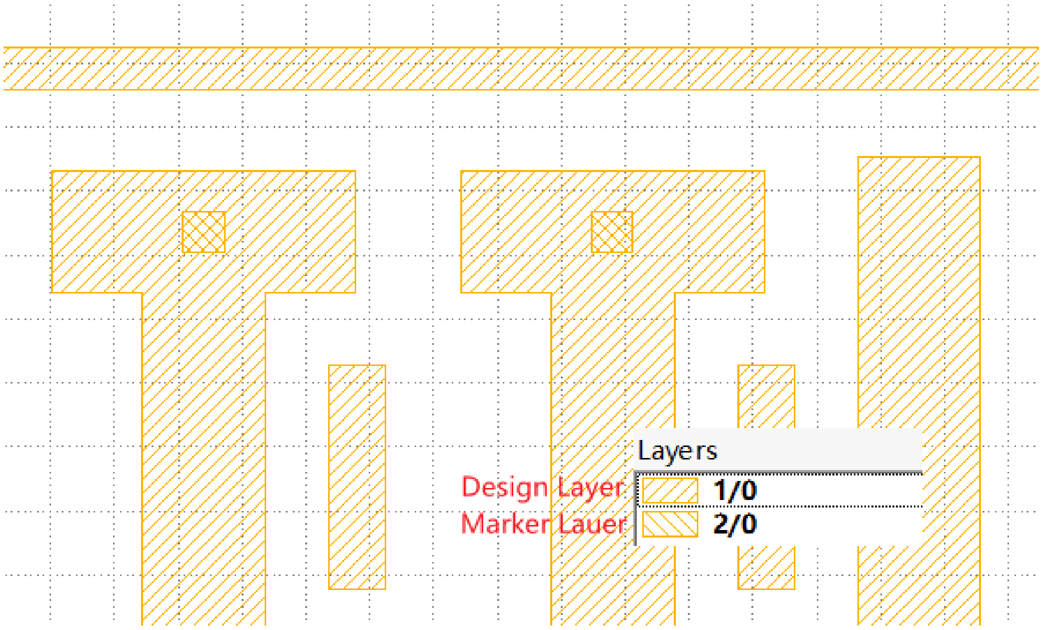


图 1 输入版图文件

2）参数配置文件：

① Pattern 半径(Pattern Radius)，所有 Pattern 形状为大小统一的正方形，第一个参数为该正方形的半径；

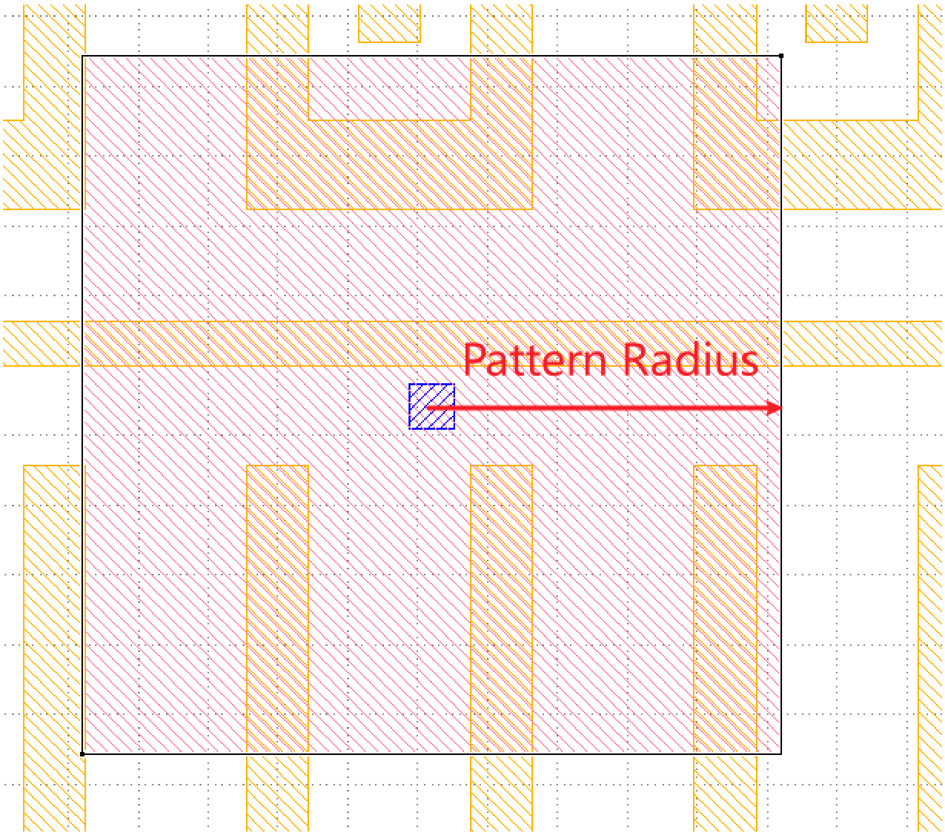


图 2 Pattern Radius

② 用例最多分类数量，要求每个提交解的分类数量要小于或等于该参数的数量；

③ 余弦相似度约束(Cosine Similarity Constraint)：

【参数范围】该数值为 0 时表示这个约束不生效；该数值生效范围为(0, 1]；

【约束值使用方法】使用该约束值前，先通过栅格化将两待比较 Pattern 由矢量图转化为位图（如图 3 所示），再使用离散余弦变换(DCT)提取频域特征向量，最后使用余弦相似度值评估两 Pattern 的相似程度，当得到的相似度值大于相似度阈值，则认为两 Pattern 可以被分到一个类簇。赛题重点考察聚类算法和 Pattern 中心点选取，因此以上栅格化、DCT 和余弦相似度等工具实现均会在答题模板中给出。

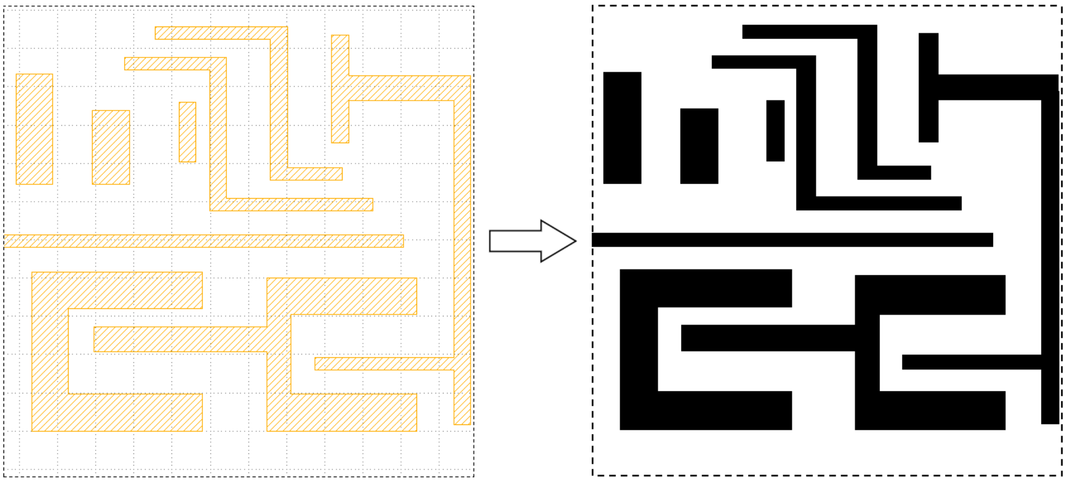


图 3 栅格化示意图

假设两个特征向量分别为 A 和 B，则余弦相似度计算公式如下：

$$\mathrm{sim}(A,B)=\frac{A\times B}{|A|\times|B|}=\frac{\sum_{i=1}^{n}A_i\times B_i}{\sqrt{\sum_{i=1}^{n}A_i^2}\times\sqrt{\sum_{i=1}^{n}B_i^2}}$$

④ 边偏移约束(Edge Movement Constraint)：

【参数范围】约束值为 0 时表示约束不生效；约束值生效范围为(0, +∞)；

【Pattern 比较前提】当判断两个 Pattern 是否可分到一个类别时，允许两 Pattern 的图形数量不一致，但图形数量较少的 Pattern 的每个图形必须要与另一个 Pattern 中的仅一个图形存在重叠；

【约束值使用方法】在 Pattern 比较过程中，允许两个对应边存在一定的偏移量，该偏移量上限由第三个参数决定。

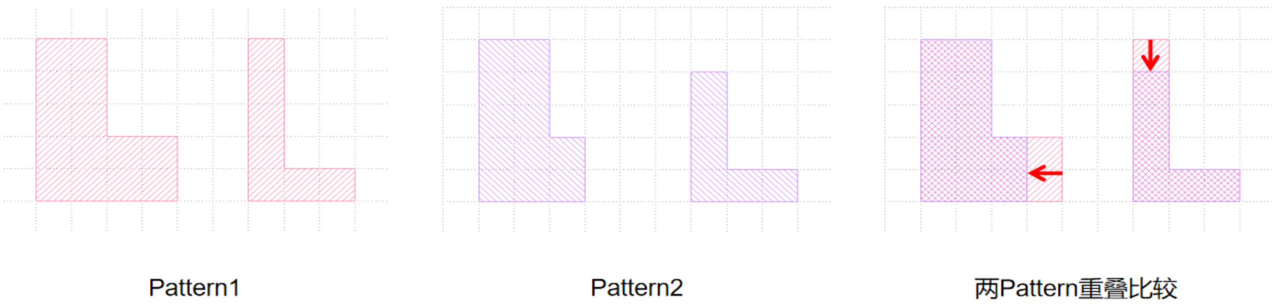


图 4 Edge Movement Constraint

如图 5、图 6 所示，余弦相似度约束和边偏移约束为两种分类场景，每个用例仅涉及其中一种场景。

case1_paras.txt

```
1500
50
0.95
0
```

图 5 余弦相似度约束用例参数配置文件

case2_paras.txt

```
1500
40
-1
5
```

图 6 边偏移约束用例参数配置文件

2. 输出文件描述：

1）Pattern 中心点位置文件：参赛者需要从每个 Marker 中选取一个点作为 Pattern 的中心点。本赛题用例中若参赛者默认选择 Marker 中心为 Pattern 中心，则仅可以获得次优解；

pattern_center_points.log

```
160110,153171
160110,157267
160110,161363
160110,165459
160110,169555
164206,153171
164206,157267
164206,161363
164206,165459
164206,169555
168302,153171
168302,157267
168302,161363
168302,165459
168302,169555
172398,153171
172398,157267
172398,161363
172398,165459
172398,169555
176494,153171
176494,157267
176494,161363
176494,165459
176494,169555
180590,153171
180590,157267
180590,161363
180590,165459
180590,169555
184686,153171
184686,157267
184686,161363
```

图 7 Pattern 中心点选取文件

2）聚类结果文件：标记根据其位置从下到上、从左到右进行标记（如图 8 所示）。参赛者应根据图案中心，根据相应的图案内容进行分类。此输出文件的第一行表示类簇的最终数量。从第二行开始，每一行都包含同一聚类标记的 ID。每行的第一个标记 ID 是该簇的簇中心。如图 8 所示，共有 5 个类，类中心分别为 7、3、12、13 和 14。只有簇中心后面的标记 ID 属于与此中心相同的簇。

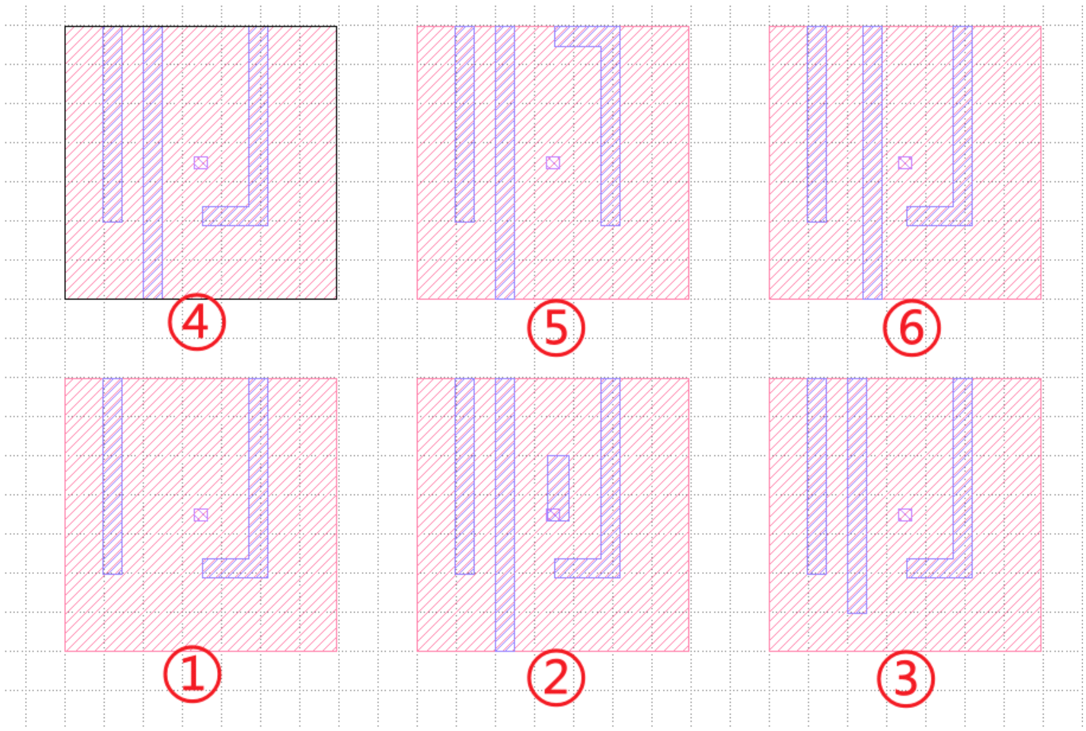
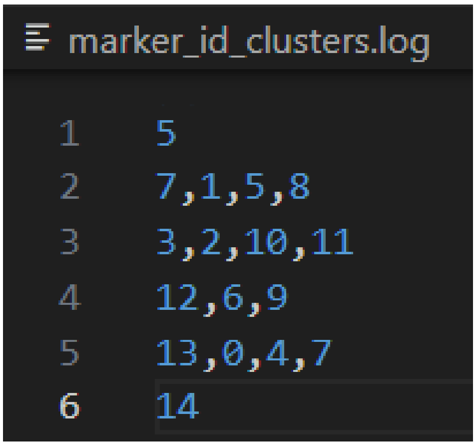


图 8 Marker 排序



```
marker_id_clusters.log
1  5
2  7,1,5,8
3  3,2,10,11
4  12,6,9
5  13,0,4,7
6  14
```

图 9 聚类结果文件

3. 运行语法：

pattern_cluster -layout./case.oas-param ./param.txt [-thread n] -pattern_centers ./pattern_center_points.txt -clusters ./marker_id_clusters.txt

pattern_cluster：图形分类可执行程序，由参赛者在大赛服务器上编译而来；

-layout：必选参数，指定版图文件的路径，该文件由出题方提供；

-param：必选参数，指定图形分类配置参数文件的路径，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，最终分数统计使用 32 线程为准；

-pattern_centers：必选参数，指定输出 pattern 中心点的位置文件的路径，由程序运行产生；

-clusters：必选参数，指定输出的分类信息的文件的路径，由程序运行产生。

4. 提交内容：

参赛队伍完成赛题后需提交以下内容：

1）编译好的可执行文件及相关静态或动态库文件；

2）所有编程源代码；

3）算法实现中涉及的三方库说明。

五、评分标准

1. 评分描述：

1）精度要求，不符合以下要求，则视为算法无效：

- 每一个 Marker 都被分类到一个类簇(Cluster)中；
- 同一个 Marker 没有被重复分类到不同类簇中；
- 所有类簇包含的 Marker 总数与版图中实际 Marker 总数一致；
- 分类的类别数量小于用例参数配置的第二个参数值（类簇数量基线）；
- 约束因素精度校验，被分到同一组的 Marker 选取的 Pattern 内容，均要与该组的类中心所选取的 Pattern 内容严格符合用例中的余弦相似度约束或者边移动约束（无需组内两两满足约束）。

以下为 4 种错误的输出内容示例：

a. 如下图所示,编号 1 的 Pattern 没有被分到某个类簇中的情况视为精度不合格,算法无效。

case1_shift_clusters.txt M ×

```
8
0,2
3,4
5,6
8,7,9
11
12,10,13,14
15,16,18
17,19
```

图 10 错误示例 1

b. 如下图所示,编号 1 的 Pattern 被重复分到两个类簇中的情况视为精度不合格,算法无效。

case1_shift_clusters.txt M ×

```
8
0,1,2
3,1,4
5,6
8,7,9
11
12,10,13,14
15,16,18
17,19
```

图 11 错误示例 2

c. 如下图所示，类簇数量为 9 组，大于用例参数配置的第二行类簇数量基线，此情况视为精度不合格，算法无效。

case1_shift_param.txt ×

```
1   300
2   8
3   0.8
4   0
```

图 12 用例参数配置

case1_shift_clusters.txt M ×

```
9
0,1,2
3,4
5,6
8,7,9
11
12,10,13,14
15,16,18
17
19
```

图 13 错误示例 3

d. 若某类簇中的 Pattern 不满足约束因素精度校验则视为精度不合格,算法无效。比如，错误示例 3 中编号 1 的 Pattern 与该组类中心（编号 0 的 Pattern）无法通过余弦相似度约束或者边移动约束的校验则该类簇分类精度不合格，算法无效。

2）提交解中包含的类簇数量越少，则分数越高；

3）算法并行耗时越少，则分数越高（以 32 线程并行为准，评分时进行绑核测试）；

4）如果算法的内存峰值(HWM)小于 64G，参赛者将获得所有 HWM 分数；

5）参赛者在打榜过程中获得的分数计算规则：

- 依据 10 次运行所得的分数，去除最高分最低分后取平均值；
- 公开用例分数在截止提交后将不再刷新；
- 相同用例多次运行得到的解要求稳定一致，若最终得分受解不稳定影响，不接受申诉重新打分。

2. 评分细则：

总分 100 分，共 4 部分：精度总分 60 分，类簇数量总分 24 分，并行性能总分 10 分，峰值 HWM 总分 6 分。每个用例仅当精度分数为满分时，其他分数项才会生效。

1）**用例分布：**根据不同规模和场景提供 6 个用例

用例	说明
Case1	小规模公开用例，余弦相似度限制场景
Case2	小规模公开用例，边偏移限制场景
Case3	大规模公开用例，余弦相似度限制场景
Case4	大规模公开用例，边偏移限制场景
Case5	大规模隐藏用例，余弦相似度限制场景
Case6	大规模隐藏用例，边偏移限制场景

2）精度得分(共 60 分)：总分被平均分配给每个用例，总精度分=通过用例数 / 总用例数 * 60；

3）**类簇数量得分(共 24 分)：**Case1 到 Case2 每个 2 分，Case3 到 Case6 每个 5 分。每个用例的最终得分占比 S 通过以下公式计算，类簇数量基线为 x_{base}，提交解类簇数量为 x（当 x 值小于 x_{base} 时才开始得分，且该项分数因计算公式局限性无法获得全部分数）；

4）**并行性能得分(共 10 分)：**Case1 到 Case2 每个 1 分，Case3 到 Case6 每个 2 分。每个用例的最终得分占比 S 通过以下公式计算，并行耗时基线为 t_{base}，提交解并行耗时为 t （由于端到端耗时中存在固定的读入写出耗时，性能优化倍数存在上限。本赛题鼓励参赛者优先获取用例的最优解，因此相同解的情况下性能优者得高分）；

5）**HWM 得分(共 6 分)：**Case3 到 Case6 每个 1.5 分，若内存峰值小于 64G，则获取对应用例全部的 HWM 分数。

六、参考资料

[1] W.-C. Chang, I. H.-R. Jiang, Y.-T. Yu, and W.-F. Liu, "iClaire: A fast and general layout pattern classification algorithm," in ACM/IEEE Design Automation Conference (DAC), 2017, pp. 64:1–64:6.DOI: 10.1145/3287624.3288747

[2] X. He, Y. Wang, Z.-Y Fu, Y.-P Wang, and Y. Guo, "A general layout pattern clustering using geometric matching-based clip relocation and lower-bound aided optimization," in ACM Transactions on Design Automation of Electronic Systems 28,2023, pp.90:1–90:23.DOI: 10.1145/3610293.

[3] K.-J. Chen, Y.-K. Chuang, B.-Y. Yu, and S.-Y. Fang, "Minimizing cluster number with clip shifting in hotspot pattern classification," in Proceedings of the 54th Annual Design Automation Conference (DAC),2017, pp. 63:1–63:6.DOI: 10.1145/3061639.3062283.

[4] C.-S. Lin, P.-Y. Tsai, Y.-H. Liu, Y.-T. Li, Y.-C. Chen and C.-Y. Wang, "Layout Hotspot Pattern Clustering Using a Density-based Approach," 2023 International VLSI Symposium on Technology, Systems and Applications (VLSI-TSA/VLSI-DAT), HsinChu, Taiwan, 2023, pp.1–4. DOI:10.1109/VLSI-TSA/VLSI-DAT57221.2023.10133968

[5] T. Vayer, L. Chapel, R. Flamary, R. Tavenard, and N. Courty, "Fused Gromov-Wasserstein distance for structured objects: theoretical foundations and mathematical properties," in ArXiv, 2018. DOI: 10.48550/arXiv.1811.02834

[6] W. C. J. Tam and R. D. S. Blanton, "LASIC: Layout analysis for systematic IC-defect identification using clustering,"in IEEE Transactions on Computer-Aided Design of Integrated Circuits Systems 34,pp.8:1278-8:1290.DOI:10.1109/TCAD.2015.2406854.

*本赛题指南未尽问题，见赛题 Q&A 文件

2025 China Postgraduate IC Innovation Competition・EDA Elite Challenge Contest

1. Problem

Very Large Scale Layout Pattern Cluster Algorithm

2. Company

HiSilicon Technologies CO., LIMITED

3. Problem Background

During the iterations of chip design and manipulation, plenty of graphic patterns which may cause defects of chips are recorded for analysis performance and yield rate. With the evolution of Very Large Scale Integration (VLSI) processes, the number of patterns typically reaches millions to billions, and these patterns often contain a large number of strictly identical or highly similar patterns. Pattern Cluster technology can significantly reduce the order of magnitude of patterns and prioritize the processing of patterns with high repetition rates.

In recent years, Pattern Cluster technology has been widely applied in multiple fields such as optical proximity correction of repetitive graphics, layout scene coverage analysis, and design rule generation. Pattern Cluster has multiple values in EDA software, including improving the efficiency of complex layout analysis, accelerating the design and verification process, and exploring potential design innovation points. Efficient and high-precision

Pattern Cluster algorithms can be the core highlight of EDA software, enhancing its competitiveness in the industry.

4. Competition Problem Description

The goal of Pattern Cluster is to extract potential hotspots in the layout, and classify these hotspots through techniques such as feature extraction or rule comparison. This competition focuses on the accuracy, performance, and peak memory indicators of Pattern Cluster algorithm implementation.

4.1 Description of Input Files:

1) Layout File (gds or oasis format), which contains two layers, as shown in Figure 1:

① Design Layer, the number and datatype of this layer is fixed at (1,0), the design layer contains a large number of Manhattan shapes, which means that the edges of the shapes must be parallel to the x-axis or y-axis;

② Marker Layer, the number and datatype of this layer is fixed at (2,0), the marker layer contains several rectangles. Each rectangle represent the selection range of pattern centers.

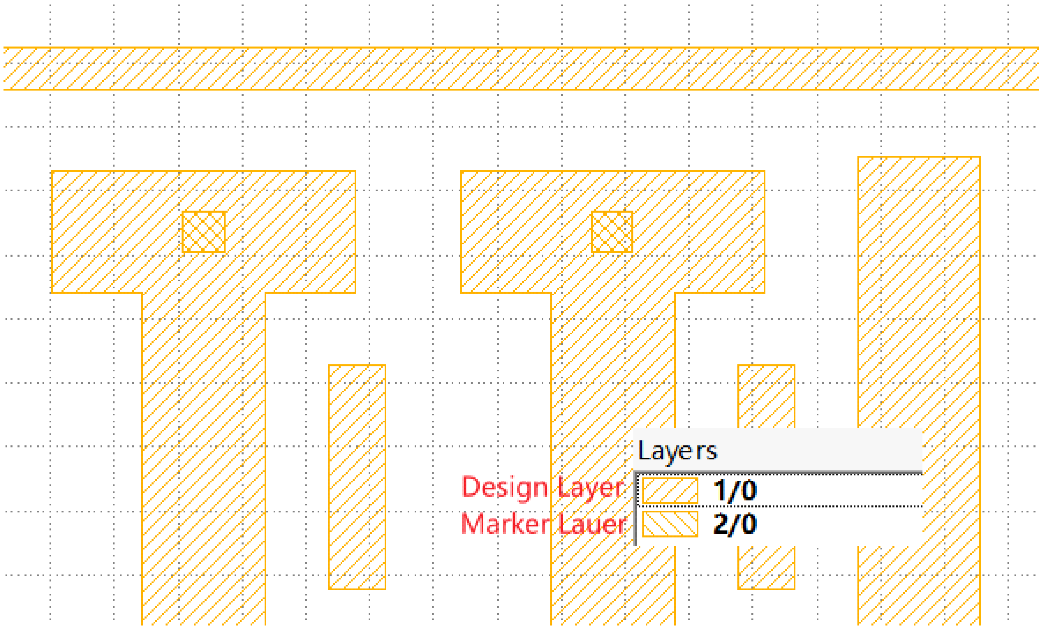


Figure 1 Input Layout file

2) Parameter configuration file:

① Pattern Radius, the Pattern shape is a square with uniform size, the first parameter is the radius of the square;

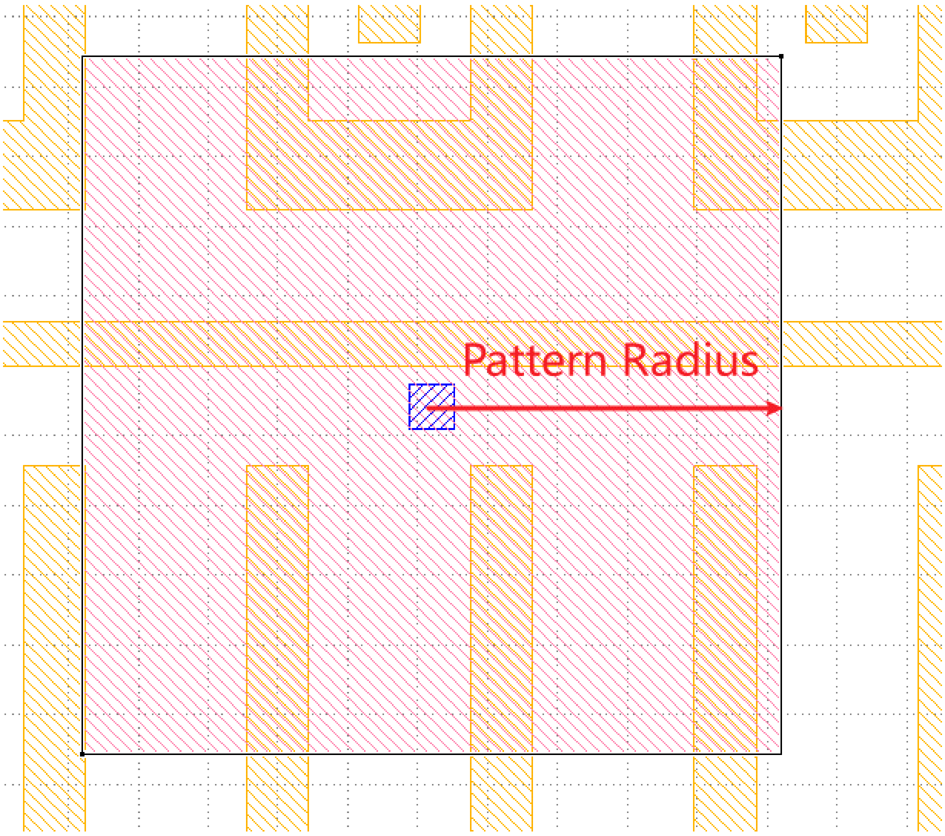


Figure 2 Pattern Radius

② The maximum number of clusters: user cases requires that the number of clusters for each submitted solution should be less than or equal to the number of the second parameter;

③ Cosine Similarity Constraint:

[Range] When the value is 0, it indicates that this constraint is not effective; The effective range of this value is (0,1].

[Method of using constraint values] Before using this constraint value, the two compared patterns are first transformed from vector images to bitmap images through rasterization (as shown in Figure 3), and then the frequency domain feature vectors are extracted using discrete cosine transform (DCT). Finally, the cosine similarity value is used to evaluate the similarity between the two patterns. When the obtained similarity value is greater than the similarity threshold, it is considered that the two patterns can be classified into a cluster. The focus of the competition is on clustering algorithms and pattern center point selection, therefore the implementation of tools such as rasterization, DCT, and cosine similarity will be provided in the answer template.

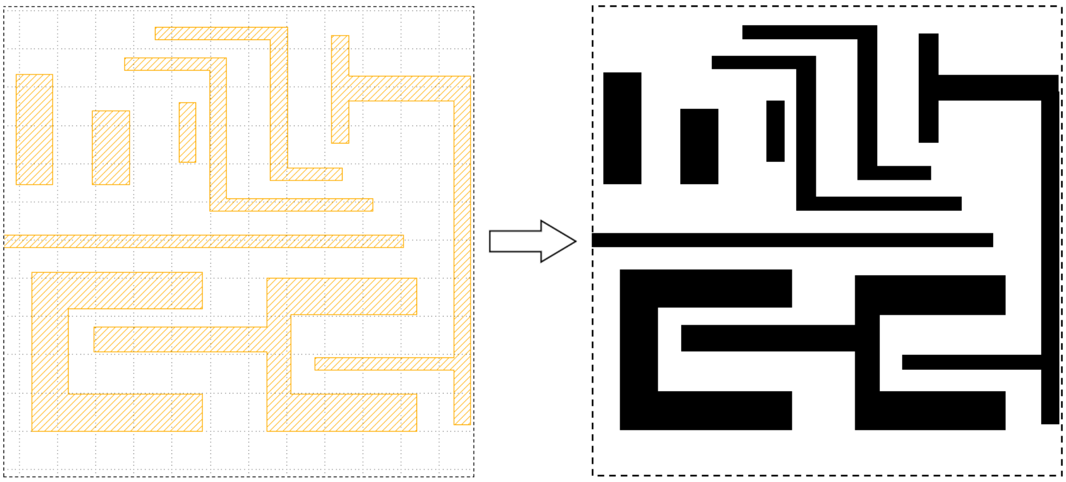


Figure 3 Rasterization

Assuming two eigenvectors are A and B, the cosine similarity calculation formula is as follows: $\mathrm{sim}(A,B)=\frac{A\times B}{|A|\times|B|}=\frac{\sum_{i=1}^{n} A_i\times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\times\sqrt{\sum_{i=1}^{n} B_i^2}}$

④ Edge Movement Constraint:

[Range] When the constraint value is 0, it indicates that the constraint is not effective; The effective range of constraint values is (0, +∞);

[Pattern Comparing premises] When determining whether two patterns can be classified into one cluster, it is allowed for the number of patterns in the two patterns to be inconsistent, but each polygon in the pattern with a smaller number of polygons must overlap with only one polygon in the other pattern;

[Method of using constraint values] During the pattern comparison process, it is allowed for two corresponding edges to have a certain movement, and the upper limit of this movement is determined by the third parameter.

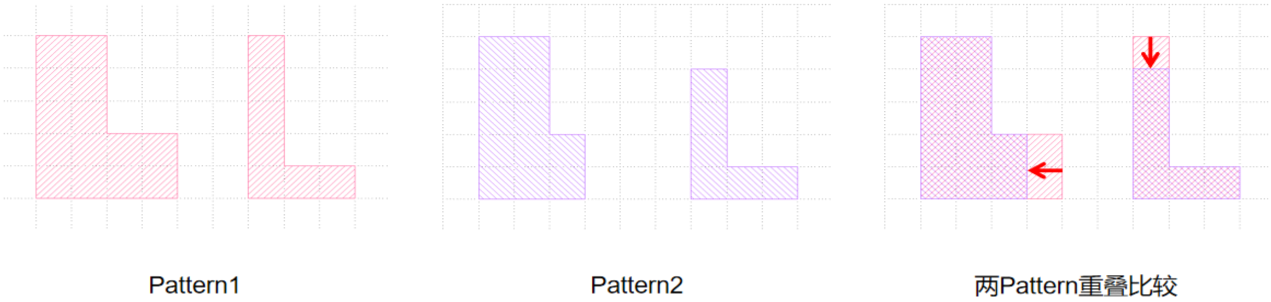
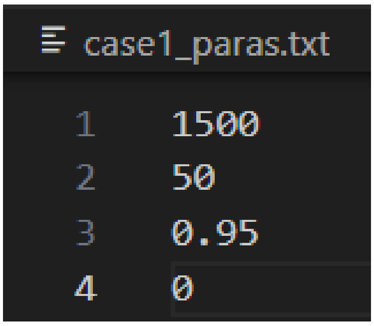


Figure 4 Edge Movement Constraint

As shown in Figures 5 and 6, the cosine similarity constraint and edge movement constraint are two cluster scenarios, and each user case only involves one of them.



case1_paras.txt

```
1500
50
0.95
0
```

Figure 5 Cosine Similarity constraint user case parameter configuration file case2_paras.txt

```
1500
40
-1
5
```

Figure 6 Edge movement constraint case parameter configuration file

4.2 Description of Input Files:

1) Pattern Centers' locations info file: Contestants need to select one point from each marker as the Pattern Center. If the participant defaults to choosing the Marker center as the Pattern center in the user cases of this competition, only sub-optimal solutions can be obtained;

pattern_center_points.log

```
160110,153171
160110,157267
160110,161363
160110,165459
160110,169555
164206,153171
164206,157267
164206,161363
164206,165459
164206,169555
168302,153171
168302,157267
168302,161363
168302,165459
168302,169555
172398,153171
172398,157267
172398,161363
172398,165459
172398,169555
176494,153171
176494,157267
176494,161363
176494,165459
176494,169555
180590,153171
180590,157267
180590,161363
180590,165459
180590,169555
184686,153171
184686,157267
184686,161363
```

Figure 7 Pattern Centers' locations info file

2) Cluster Result File: Markers are labeled from bottom to top, and from left to right based on their position (as shown in Figure 8). Contestants should classify them according to the corresponding pattern content based on the pattern centers. The first line of this output file represents the final number of clusters. Starting from the second line, each line contains the ID of the same cluster marker. And the first marker ID of each line is the cluster center of that cluster. As shown in Figure 8, there are a total of 5 classes, with class centers of 7, 3, 12, 13, and 14 respectively. Only the markers' ID following the cluster center belong to the same cluster with this center.

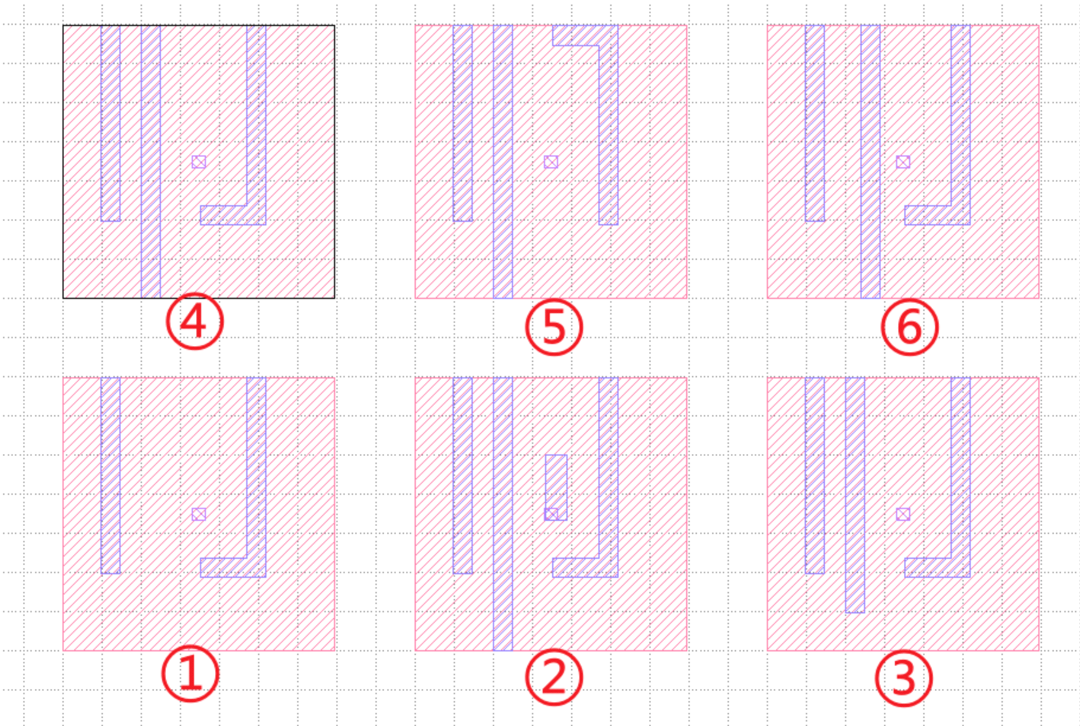


Figure 8 Marker order

marker_id_clusters.log

```
5
7,1,5,8
3,2,10,11
12,6,9
13,0,4,7
14
```

Figure 9 Cluster Result file

4.3 Command line syntax:

pattern_cluster -layout ./case.oas -param ./param.txt [-thread n] -pattern_centers ./pattern_center_points.txt -clusters ./marker_id_clusters.txt

pattern_cluster: Pattern cluster executable program, compiled by contestants on the competition server;

-layout: required parameter, which specifies the path of the layout file, which is provided by the problem solving party;

-param: required parameter, which specifies the path of the pattern cluster configuration parameters file, which is provided by the problem solving party;

-thread: optional parameter, which specifies n threads for parallel computing. And the final score of each user cases will be collected under 32 threads;

-pattern_centers: required parameter, which specifies the path of the location file for the center point of the output pattern, generated by the program;

-clusters: required parameter, which specifies the path of the file containing the output clusters information, generated by the program.

4.4 Submit Contents:

After completing the competition problem, the participating teams need to submit the following contents:

1) Compiled executable files, and its related static or dynamic library files;
2) All programming source code;
3) Explanation of the third-party libraries involved in algorithm implementation.

5. Standard Scoring Criteria

5.1 Evaluation Standard：

1) Accuracy requirements, if the following requirements are not met, the algorithm will be considered invalid:

- Each marker should be classified into only one cluster;
- The same marker should not be repeatedly classified into different clusters;
- The total number of markers contained in all clusters should be consistent with the actual total number of markers in the layout;
- The number of clusters should be less than the second parameter value of the user cases' parameter configuration (baseline for the number of clusters);
- Accuracy verification of constraints: The pattern content selected by the Markers assigned to the same group must strictly comply with the cosine similarity constraint or edge movement constraint in the user cases with the pattern content selected by the cluster center of the group (without the need for pairwise satisfy constraints within the group).

Here are four examples of incorrect output content:

a. As shown in the figure 1, if Pattern 1 is not assigned to a certain cluster, it is considered to have poor accuracy and the algorithm is invalid.

case1_shift_clusters.txt M ×

```
8
0,2
3,4
5,6
8,7,9
11
12,10,13,14
15,16,18
17,19
```

Figure 10 Wrong output example 1

b. As shown in Figure 2, if Pattern 1 is repeatedly assigned to two clusters, it is considered inaccurate and the algorithm is invalid.

case1_shift_clusters.txt M ×

```
8
0,1,2
3,1,4
5,6
8,7,9
11
12,10,13,14
15,16,18
17,19
```

Figure 11 Wrong output example 2

c. As shown in Figure 3, the number of clusters is 9, which is greater than the baseline number of clusters in the second row of the use case parameter configuration. This situation is considered to be inaccurate and the algorithm is invalid.

case1_shift_param.txt ×

```
300
8
0.8
0
```

Figure 12 case parameters

case1_shift_clusters.txt M ×

```
9
0,1,2
3,4
5,6
8,7,9
11
12,10,13,14
15,16,18
17
19
```

Figure 13 Wrong output example 3

d. If the patterns in a certain cluster do not meet the accuracy check of the constraint factors, it is considered that the accuracy is unqualified and the algorithm is invalid. For example, if the pattern with the number 1 in ‘Wrong output example 3’ cannot pass the cosine similarity constraint or edge movement constraint verification with the center of the group (pattern with the number 0), then the classification accuracy of the class cluster is not qualified and the algorithm is invalid.

2) The fewer clusters included in the submitted solution, the higher the score;

3) The lower the parallel processing time of the algorithm, the higher the score (based on 32 threads in parallel, with kernel binding testing conducted during scoring);

4) If the high water mark (HWM) of the algorithm is less than 64G, contestants obtain all of the HWM score;

5) The calculation rules for the scores obtained by contestants during the ranking process：

- Based on the scores obtained from 10 runs, remove the highest and lowest scores and take the average;
- The score for public user cases will no longer be refreshed after the submission

deadline;

- The solution obtained by running the same user case multiple times requires stability and consistency. If the final score is affected by unstable solutions, appeals will not be accepted for re-scoring.

5.2 Scoring Criteria:

Total score of 100 points, consisting of 4 parts: The total score for accuracy is 60 points, the total score for the number of clusters is 24 points, the total score for parallel performance is 10 points, and the total score for peak HWM is 6 points. For each user case, other score items will take effect only when the accuracy score is full.

1) **User case distribution:** 6 user cases are provided based on different scales and scenarios

User cases	**Explanation**
Case1	Small scale, public, cosine similarity constraint scenario
Case2	Small scale, public, edge movement constraint scenario
Case3	Large scale, public, cosine similarity constraint scenario
Case4	Large scale, public, edge movement constraint scenario
Case5	Large scale, hidden, cosine similarity constraint scenario
Case6	Large scale, hidden, edge movement constraint

	scenario

2) **Accuracy score (60 points in total):** The total score is evenly distributed to each user case, final accuracy score=number of pass cases / number of all cases * 60;

3) **The number of clusters score (24 points in total):** Case1 and Case2 respectively have 2 points, and each of Case3 to Case6 has 5 points. The final score proportion S for each user case is calculated using the following formula, the baseline for the number of clusters is x_{base}, the number of submitted solution clusters is x (Scoring only starts when the value of x is less than x_{base}, and due to the limitations of this calculation formula, it is not possible to obtain all scores):

$$S = \left(\frac{x_{base} - x}{x_{base}}\right) * 100\%$$

4) **Parallel performance score (10 points in total)：** Case1 and Case2 respectively have 1 point, and each of Case3 to Case6 has 2 points. The final score proportion S for each user case is calculated using the following formula, the baseline for parallel performance is t_{base}, the submission parallel performance is t (Due to the fixed read and write time in end-to-end processing, there is an upper limit to the performance optimization factor. This competition encourages participants to prioritize obtaining the optimal solution of the user case. Therefore, in the case of the same solution, the one with better performance will receive high scores):

$$\begin{cases} S = \left(\frac{x_{base} - x}{x_{base}}\right) * \left(\frac{{t_{base}}^2}{t^2 * 10^4}\right) * 100\%, \frac{t_{base}}{t} \leq 100; \\ S = 100\%, \frac{t_{base}}{t} > 100; \end{cases}$$

5) **Peak HWM Score(6 points in total)：** Each of Case3 to Case6 has 1.5 points, for each user case, if the peak memory is less than 64G, obtain all of the Peak HWM scores of corresponding case.

6. References

[7] W.-C. Chang, I. H.-R. Jiang, Y.-T. Yu, and W.-F. Liu, "iClaire: A fast and general layout pattern classification algorithm," in ACM/IEEE Design Automation Conference (DAC), 2017, pp. 64:1–64:6.DOI: 10.1145/3287624.3288747

[8] X. He, Y. Wang, Z.-Y Fu, Y.-P Wang, and Y. Guo, "A general layout pattern clustering using geometric matching-based clip relocation and lower-bound aided optimization," in ACM Transactions on Design Automation of Electronic Systems 28,2023, pp.90:1–90:23.DOI: 10.1145/3610293.

[9] K.-J. Chen, Y.-K. Chuang, B.-Y. Yu, and S.-Y. Fang, "Minimizing cluster number with clip shifting in hotspot pattern classification," in Proceedings of the 54th Annual Design Automation Conference (DAC),2017, pp. 63:1–63:6.DOI: 10.1145/3061639.3062283.

[10] C.-S. Lin, P.-Y. Tsai, Y.-H. Liu, Y.-T. Li, Y.-C. Chen and C.-Y. Wang, "Layout Hotspot Pattern Clustering Using a Density-based Approach," 2023 International VLSI Symposium on Technology, Systems and Applications (VLSI-TSA/VLSI-DAT), HsinChu, Taiwan, 2023, pp.1–4. DOI:10.1109/VLSI-TSA/VLSI-DAT57221.2023.10133968

[11] T. Vayer, L. Chapel, R. Flamary, R. Tavenard, and N. Courty, "Fused Gromov-Wasserstein distance for structured objects: theoretical foundations and mathematical properties," in ArXiv, 2018. DOI: 10.48550/arXiv.1811.02834

[12] W. C. J. Tam and R. D. S. Blanton, "LASIC: Layout analysis for systematic IC-defect

identification using clustering,"in IEEE Transactions on Computer-Aided Design of Integrated Circuits Systems 34,pp.8:1278-8:1290.DOI:10.1109/TCAD.2015.2406854.

*For questions not covered in this guide, please refer to the O&A document