2025 中国研究生创“芯”大赛·EDA 精英挑战赛

一、赛题名称

版图多层链路追踪算法

二、命题单位

杭州广立微电子股份有限公司

三、赛题背景

芯片版图链路追踪（Net Trace）技术是芯片设计与制造中的关键技术之一，目前已成为提升芯片良率、降低设计风险的重要手段。该技术在芯片设计和制造的应用场景包括：

1）芯片设计阶段连接性验证与一致性检查：Net trace 技术通过追踪版图中的物理连线，确保实际版图与逻辑原理图（如 SPICE 网表）的连接关系完全一致。例如，在 LVS[1]（版图与原理图一致性检查）过程中，该技术可识别短路、断路或信号路径偏差，避免因设计错误导致的功能失效；

2）制造缺陷定位与良率提升[2]：通过追踪晶圆测试中的失效点，结合版图数据，Net trace 可快速定位工艺波动或光刻异常导致的物理缺陷（如金属桥接或孔洞），优化工艺参数；

3）设计优化：如寄生参数提取流程中通过 Net trace 技术追踪金属连线的几何形状和材料特性[3]，提取寄生电阻、电容和电感（RC Extraction），用于电路仿真。在功耗与散热优化流程[4]中通过 Net trace 技术分析电源网络（Power Grid）的电流密度和压降（IR Drop），避免局部过热或供电不足。

在先进工艺节点下，随着芯片规模和复杂度的指数级增长，版图链路追踪技术面

临数据量激增、多层连接性关系复杂度提升等多重挑战，推动着 EDA 工具和算法的持续创新。

版图链路追踪技术属于数字/模拟/射频版图的物理验证领域，主要应用在前述的芯片设计验证，优化和制造良率提升阶段。本赛题将模拟工业界真实的版图链路追踪场景，输入为待分析版图，版图层级连接性关系和追踪起点，输出为追踪全路径结果。

四、赛题描述

链路追踪的最基础应用为输入起点层，起点坐标和层之间的连接关系，在版图文件中搜索起点所在多边形的连通区域，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 个规则文件。

1）规则文件：由起点、Via 规则和 Gate 规则组成。本赛题用文本文件来描述规则文件，其格式如图 1 所示。其包含了两部分，第一部分描述了规则文件中包含的起点的信息。以 StartPos 开始（第 1 行）。紧随的一行描述了起点的层（L2）及坐标，表示链路追踪的起点从 L2 层的(-65,31)这个位置开始。第 3 行进入第二部分，描述了 Via 规则的信息，以 Via 开头，紧随其后的第 4 行描述了 L1、L2 和 L3 的 Via 关系，表示：**L1 通过 L2 与 L3 相连，即 L1 与 L2 上彼此相交的多边形视为同一个连通区域，L2 与 L3 上彼此相交的多边形视为同一个连通区域**。

```
1 StartPos
2 L2 (-65,31)
3 Via
4 L1 L2 L3
```

图 1a

```
1 StartPos
2 L4 (100,50)
3 Via
4 L1 L3 L4
5 L2 L3 L4
```

图 1b

请注意以下几点：

a) 每个规则文件可以有一个或两个起点、一个或多个 Via 规则和至多一个 Gate

规则；

b) 起点的坐标值为 32 位有符号整数；

c) Via 规则在每个规则文件中可能有一个或多个。Via 规则中每个 Layer 仅包含字母数字和下划线，每个 Via 规则可以有至少 1 个 Layer，每个 Via 规则中不含重复的 Layer。**包含多个 Layer 的 Via 规则中相邻的 Layer 上彼此相交的多边形视为同一个连通区域，不相邻的 Layer 间则无连接性关系。**如图 1a 的单个 Via 规则例子，L1 与 L2 或 L2 与 L3 相交的图形视为相互连通，但 L1 与 L3 相交的图形则视为相互不连通。如图 1b 的两个 Via 规则例子，L1 与 L3，L3 与 L4, L2 与 L3 层的接触区域都存在连通关系，但 L1 与 L4，L2 与 L4 或 L1 与 L2 之间不能直接连通。如多条 Via 规则中存在重复定义的连接性关系，则忽略（如图 1b 例子中的 L3 与 L4）。

d) Gate 规则在每个规则文件中可能有一个或没有，其作用是指定芯片版图中的 Poly（多晶硅）和 AA（有源区）层。该规则仅包含两层，第一层即指定为 Poly 层，第二层指定为 AA 层。在芯片版图中，如果 Poly 层被连接到高电平信号，则与之相交的 AA 两侧区域（源极 Source 和漏极 Drain）将处于连通状态；反之如果 Poly 层被连接到低电平信号，则与之相交的 AA 两侧区域将处于断开状态。如图 2a，如果 P1 被连接到高电平信号，P2 被连接到低电平信号，则 S1 和 D1 将处于连通状态，而 S2 和 D2 则处于断开状态。

多边形相交条件为：要么两个多边形有部分区域重叠且重叠区域面积大于 0，见图 b 中的重叠关系；要么两个多边形边界相交，见图 2b 中的边界点相交关系和边界线相交关系。**即图 b 中的三种情况（重叠、边界点相交和边界线相交）都属于多边形相交。**

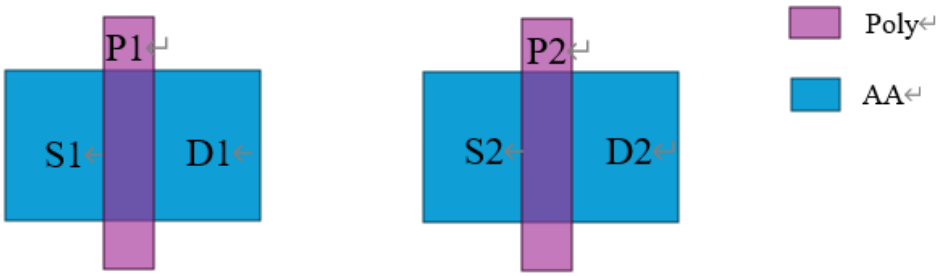

图 2a

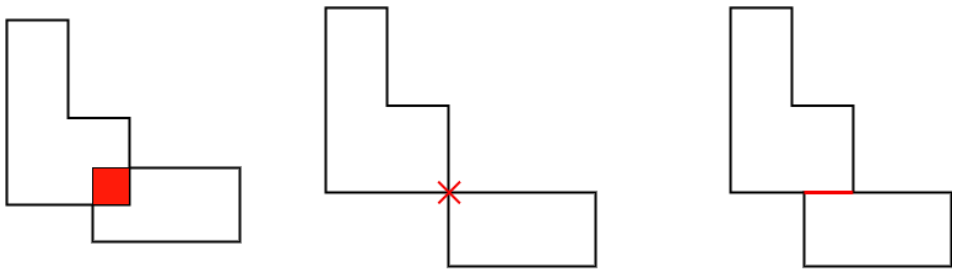

图 2b

2）版图文件：

a) 版图文件：本赛题采用文本文件，而非标准的 GDSII 或 OASIS 文件来描述版图，版图文件中包含若干层的若干个曼哈顿多边形（以下简称为多边形）组成，每层的多边形以 Layer 标识（即层名）开头，到下一个 Layer 标识或者文件结尾截止。Layer 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。其格式如图 所示，包含每个 Layer 上的多边形数据。第 1 行代表输入的第一个 Layer 标识为 L1，第 2~3 行代表 L1 上的两个多边形，第 4 行代表输入的第二个 Layer 标识为 L2，第 5 行代表 L2 上的一个多边形，第 6 行代表输入的第三个 Layer 标识为 L3，第 7~9 行代表 L3 上的三个多边形。图 3 的所有多边形的显示图可见图 4，其中版图的所有的图形显示在虚线边框的矩形区域内，矩形区域右上角提示了版图中每个 Layer 上多边形的填充颜色，图中的注解#Layer@#num 表示这个多边形为#Layer 层上的第#num 个多边形（Layer 上#num 顺序与输入版图文件的多边形顺序一致），例如 L1@1 表示 L1 层上

的第 1 个多边形。请注意以下几点：

- 输入保证版图的多边形均为曼哈顿图形（所有的边均平行或垂直于平面坐标轴）且不含内孔和自交；
- 输入的多边形的坐标值为 32 位有符号整数；
- 输入的 Layer 标识只包含字母数字和下划线；

```
L1
(-5,-15),(10,-15),(10,10),(-5,10)
(15,-4),(5,-4),(5,-22),(23,-22),(23,5),(15,5)
L2
(0,0),(0,38),(-13,38),(-13,15),(-36,15),(-36,0)
L3
(-49,28),(-29,28),(-29,78),(-49,78)
(-73,14),(-11,14),(-11,40),(-73,40)
(-15,-25),(10,-25),(10,-17),(-15,-17)
```

图 3

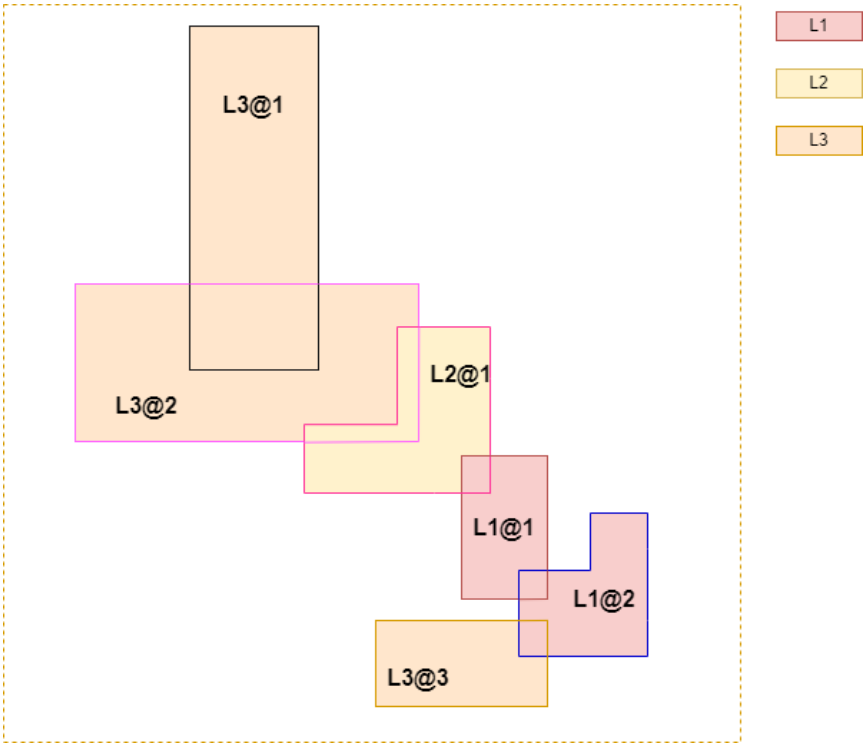

图 4

2. 链路追踪要求：

1）从规则文件中的起点开始，在版图文件中搜索图形，搜索时满足规则文件中的

Via 规则和 Gate 规则，输出满足指定要求的从指定起点开始的所有连通多边形的坐标结果。输出的多边形按照 Layer 进行分组（要求多边形的顶点按照逆时针顺序输出）。例如图 5 展示了用 via 规则（L1 L2 L3）在图 3 的版图中搜索的结果，图中红色十字代表规则文件中定义的起点位置，该位置在 L3 层第 2 个多边形内，带红色粗线条的多边形为搜索到的与起点连通的多边形，对应结果文件应为图 所示。L1 上的搜索结果为 L1@1, L1@2; L2 上的搜索结果为 L2@2, L3 上的搜索结果为 L3@1 和 L3@2，由于 L3 与 L1 两层没有连接关系（Via 关系中 L1 与 L3 不相邻），所以 L3@3 不能通过 L1@2 连接得到。

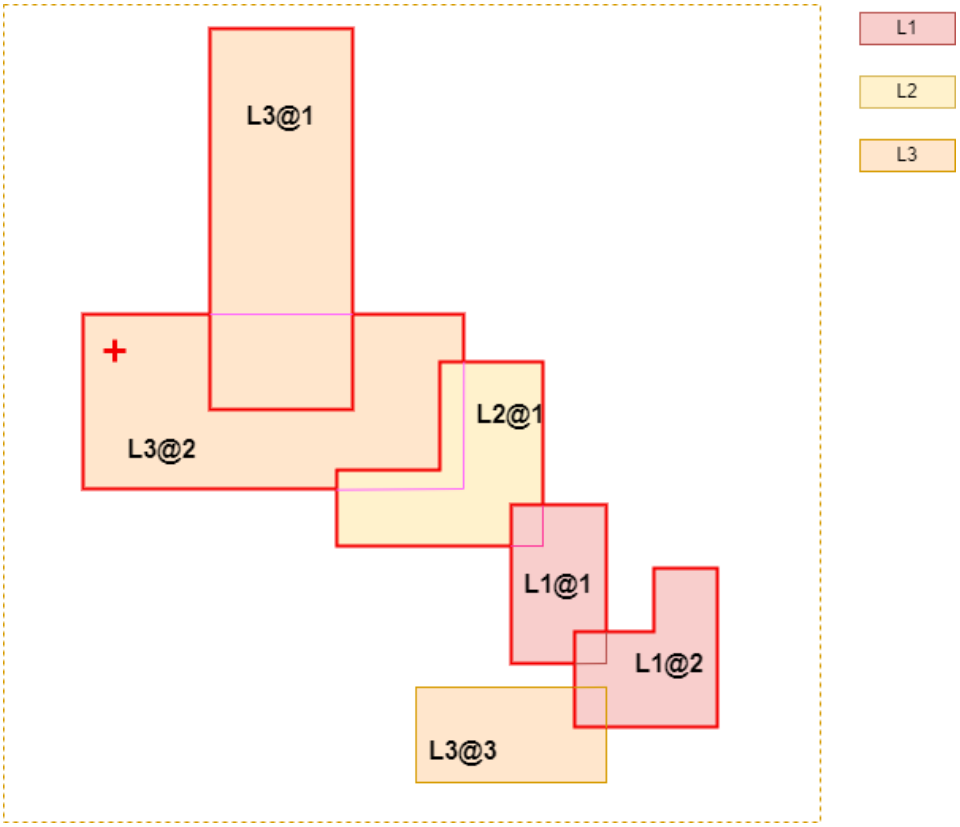

图 5

```
1 L1
2 (-5,-15),(10,-15),(10,10),(-5,10)
3 (5,-22),(23,-22),(23,5),(15,5),(15,-4),(5,-4)
4 L2
5 (0,0),(0,38),(-13,38),(-13,15),(-36,15),(-36,0)
6 L3
7 (-49,28),(-29,28),(-29,78),(-49,78)
8 (-73,14),(-11,14),(-11,40),(-73,40)
```

图 6

2）本赛题包含三个小题：

第一小题考察单个 Layer 和单个起点条件下的链路追踪，仅有一条 via rule 和一个起点位置，via rule 也仅包含一个 Layer，该 Layer 与起点的位置 Layer 相同。如图中所示的版图，起点用红色十字表示，从起点开始搜索，正确的连通多边形应该包含版图中所有的多边形。

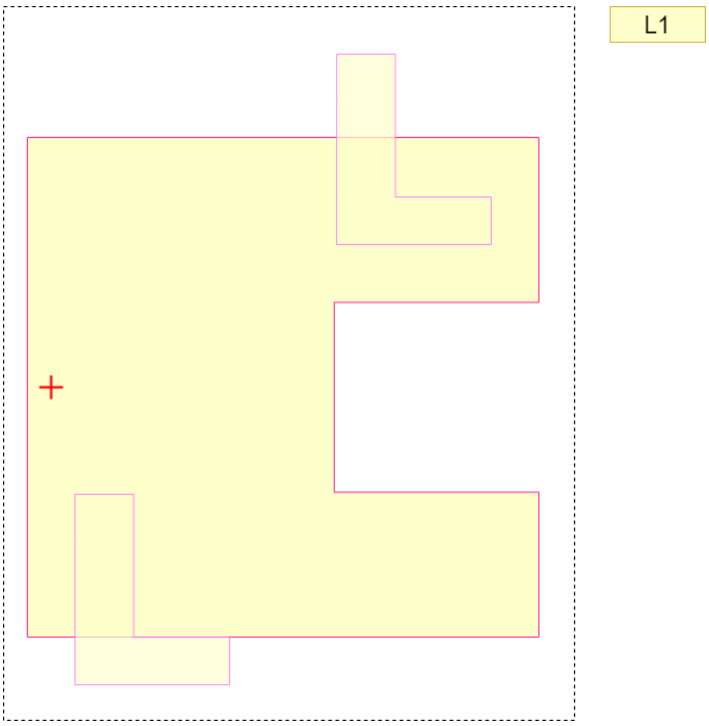

图 7

第二小题考察多个 Layer 和单个起点条件下的链路追踪，如图 中版图文件和图 1 中规则文件定义的问题，参赛选手需要考虑规则文件中的 Via 规则，在不同 Layer 之间寻找连通多边形。输出所有与起点位置连通的各层多边形的结果。

第三小题模拟在考虑芯片逻辑电路晶体管开关状态链路追踪问题。该小题将在第一个起点位置 s1 为给 Poly 输入电信号的位置起点，接入高电平信号后，则与之有连接性关系（需考虑多层的 via rule）的 Poly 将全部置为高电平信号，其余 Poly 则置为低电平信号。在此状态下，从第二个起点位置 s2 开始搜索，在考虑多层 via rule 的情况下，输出所有与 s2 位置连通的各层多边形结果。注意，**与 s1 位置连通的各层多边形结果不需要输出。**

若使用图 9 中的规则文件来对图 8 中的版图文件进行链路追踪，得到的结果为图 9 中右图的结果文件，图 9 中的结果显示图为图 10，其中黑色边框的多边形为第一个起点的连通区域，红色边框的多边形为第二个起点的连通区域。图 10 中黑色十字为第一个起点的位置，红色十字为第二个起点的位置。在该例中，存在 Gate rule，其中 PO 指定为 Poly 层，AA 指定为 AA 层。根据 via rule，M1 与 CT 存在连接性关系，CT 与 PO 存在连接性关系，所以从 M1 层的 s1 位置（118743, 438448）出发，图 10 左边的 PO 层图形获得高电平信号，与之相交的图 10 上方 AA 层图形左右两侧区域处于连通状态。而右边的 PO 层由于与 s1 不连通，因此图 10 右边 PO 层左右两侧区域处于断开状态。所以最终从 s2 出发的连通结果为图 10 中红色边框的多边形，这些多边形的坐标即为赛题要求的输出结果。请注意以下几点：

- 输入的版图文件中，若 AA 层的多边形 P1 与 Poly 层的多边形 P2 相交，则 P2 必定贯穿 P1，即 P2 必定可以将 P1 划分成两部分。
- 如果 AA 层上的多边形 P1 与 s2 连通，并且与 Poly 层的多边形 P2 相交，则需要计算 P1 被 P2 切割后的多边形，再将切割后的多边形放入结果中。例如图 10 中 AA 层上侧的多边形被 PO 层的多边形贯穿，其被 PO 层多边形切割后的多边形为 AA@1、AA@2 和 AA@3。由于贯穿 AA@1 和 AA@2 的 PO 层多边形与 s1 相连，处于高电平

状态，因此 AA@1 与 AA@2 连通。同样的，AA 层下侧的多边形被切割为 AA@4、AA@5 和 AA@6，切割 AA@5 和 AA@6 的 PO 层多边形不与 s1 相连，故 AA@5 不在 s2 的搜索结果中。

```
AA
(118870,436985),(119770,436985),(119770,437985),(118870,437985)
(118820,439200),(119720,439200),(119720,440200),(118820,440200)
PO
(119460,438155),(119460,436865),(119560,436865),(119560,438385),
(119510,438385),(119510,440310),(119410,440310),(119410,438155)
(119080,438145),(119080,436865),(119180,436865),(119180,438375),
(119130,438375),(119130,440310),(119030,440310),(119030,438375),
(118890,438375),(118890,438145)
CT
(119580,440050),(119710,440050),(119710,440180),(119580,440180)
(119200,440050),(119330,440050),(119330,440180),(119200,440180)
(118830,440050),(118960,440050),(118960,440180),(118830,440180)
(119580,439770),(119710,439770),(119710,439900),(119580,439900)
(119200,439770),(119330,439770),(119330,439900),(119200,439900)
(118830,439770),(118960,439770),(118960,439900),(118830,439900)
(119580,439490),(119710,439490),(119710,439620),(119580,439620)
(119200,439490),(119330,439490),(119330,439620),(119200,439620)
(118830,439490),(118960,439490),(118960,439620),(118830,439620)
(119580,439210),(119710,439210),(119710,439340),(119580,439340)
(119200,439210),(119330,439210),(119330,439340),(119200,439340)
(118830,439210),(118960,439210),(118960,439340),(118830,439340)
(119420,438245),(119550,438245),(119550,438375),(119420,438375)
(118900,438235),(119030,438235),(119030,438365),(118900,438365)
(119630,437845),(119760,437845),(119760,437975),(119630,437975)
(118880,437835),(119010,437835),(119010,437965),(118880,437965)
(119630,437565),(119760,437565),(119760,437695),(119630,437695)
(118880,437555),(119010,437555),(119010,437685),(118880,437685)
(119630,437285),(119760,437285),(119760,437415),(119630,437415)
(118880,437275),(119010,437275),(119010,437405),(118880,437405)
(119630,437005),(119760,437005),(119760,437135),(119630,437135)
(118880,436995),(119010,436995),(119010,437125),(118880,437125)
M1
(119580,438715),(119710,438715),(119710,441225),(119580,441225)
(118830,438715),(118960,438715),(118960,441225),(118830,441225)
(118650,438165),(119030,438165),(119030,438585),(118650,438585)
(118880,436813),(119010,436813),(119010,438035),(118880,438035)
(119420,438175),(119765,438175),(119765,438585),(119420,438585)
(119160,438715),(119160,436935),(119760,436935),(119760,437885),
(120020,437885),(120020,437815),(120150,437815),(120150,438085),
(120020,438085),(120020,438015),(119760,438015),(119760,438045),
(119630,438045),(119630,437065),(119290,437065),(119290,438715),
(119400,438715),(119400,440180),(119130,440180),(119130,438715)
```

图 8

```
StartPos
M1 (118743,438448)
M1 (118871,441132)
Via
AA CT M1
PO CT M1
Gate
PO AA
```

```
AA
(118820,439200),(119030,439200),(119030,440200),(118820,440200)
(119130,439200),(119410,439200),(119410,440200),(119130,440200)
(119560,436985),(119770,436985),(119770,437985),(119560,437985)
CT
(118830,440050),(118960,440050),(118960,440180),(118830,440180)
(118830,439770),(118960,439770),(118960,439900),(118830,439900)
(118830,439490),(118960,439490),(118960,439620),(118830,439620)
(118830,439210),(118960,439210),(118960,439340),(118830,439340)
(119200,440050),(119330,440050),(119330,440180),(119200,440180)
(119200,439770),(119330,439770),(119330,439900),(119200,439900)
(119200,439490),(119330,439490),(119330,439620),(119200,439620)
(119200,439210),(119330,439210),(119330,439340),(119200,439340)
(119630,437845),(119760,437845),(119760,437975),(119630,437975)
(119630,437565),(119760,437565),(119760,437695),(119630,437695)
(119630,437285),(119760,437285),(119760,437415),(119630,437415)
(119630,437005),(119760,437005),(119760,437135),(119630,437135)
M1
(118830,438715),(118960,438715),(118960,441225),(118830,441225)
(119160,438715),(119160,436935),(119760,436935),(119760,437885),
(120020,437885),(120020,437815),(120150,437815),(120150,438085),
(120020,438085),(120020,438015),(119760,438015),(119760,438045),
(119630,438045),(119630,437065),(119290,437065),(119290,438715),
(119400,438715),(119400,440180),(119130,440180),(119130,438715)
```

图 9 左图为规则文件，右图为结果文件

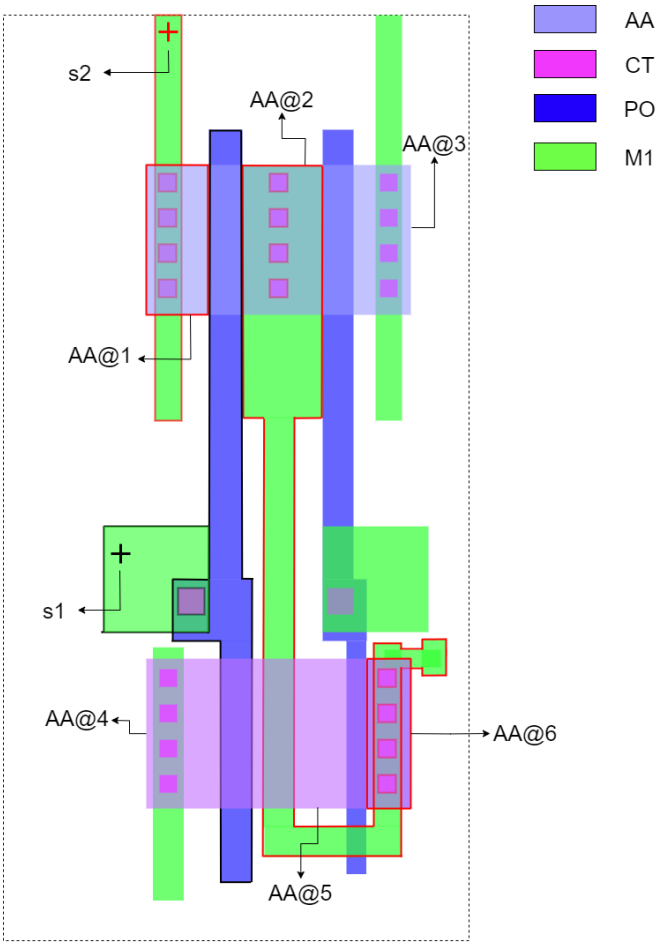

图 10

3. 输出文件格式：

输出文件格式为 linux 系统的文本文件格式。内容为当前规则文件中起点在当前版图文件搜索到的所有结果的合集，按照 Layer 对多边形分组每个多边形占一行，结果的记录方式参见链路追踪要求章节。

4. 运行语法：

trace -layout ./layout.txt -rule ./rule.txt [-thread n] -output ./res.txt

trace：链路追踪（Net Trace）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。

-rule: 必选参数，指定规则文件的路径，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

五、 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。本赛题共三个小题，每个小题完全正确，无错误或遗漏的多边形，则得 20 分。每有 1 个错误多边形或遗漏，扣 1 分，扣完为止。将 3 个搜索结果的得分相加，所得分数即为正确性得分。

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。

单线程性能分按照 Trace 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照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，则该项得分记为 0 分。**

多线程性能分按照 Trace 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照从小到大的顺序进行排序，第一名得 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。

六、 参考资料

1. Pandey R, Agrawal N, Arghavani R, et al. Analysis of local interconnect resistance at scaled process nodes[C]//2015 73rd Annual Device Research Conference (DRC). IEEE, 2015.DOI:10.1109/DRC.2015.7175620.

2. 余建波, 卢笑蕾, 宗卫周.基于局部与非局部线性判别分析和高斯混合模型动态集成的晶圆表面缺陷探测与识别[J]. 自动化学报, 2016, 42(1):13.DOI:CNKI:SUN:MOTO.0.2016-01-005.

3. Chen D C, Lin G S, Lee T H, et al. Compact modeling solution of layout dependent effect for FinFET technology[C]//2015 International Conference on Microelectronic Test Structures (ICMTS).IEEE, 2015.DOI:10.1109/ICMTS.2015.7106119.

4. Pandey R, Agrawal N, Arghavani R, et al. Analysis of local interconnect resistance at scaled process nodes[C]//2015 73rd Annual Device Research Conference (DRC). IEEE, 2015.DOI:10.1109/DRC.2015.7175620.

*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

Layout Net Tracing Algorithm

2. Company

Semitronix Corporation

3. Problem Background

The chip layout net tracing (Net Trace) technology is one of the key techniques in integrated circuit (IC) design and manufacturing and has become an important means of improving chip yield and reducing design risk. Its applications in IC design and manufacturing include:

1) Connectivity verification and consistency checking during the design stage: Net trace technology traces the physical wires in the layout to ensure that the actual layout is fully consistent with the logical schematic (e.g., a SPICE netlist). For example, in Layout Versus Schematic (LVS)[1], it can identify shorts, opens, or deviations in signal paths, thereby preventing functional failures caused by design errors.

2) Manufacturing defect localization and yield improvement[2]: By tracing failure points observed in wafer tests and correlating them with layout data, net trace can rapidly locate physical defects—such as metal bridging or voids—caused by process variations or lithographic anomalies, and can be used to optimize process parameters.

3) Design optimization: In parasitic extraction flows, net trace technology traces the geometric shapes and material properties of metal interconnects[3] to extract parasitic

resistance, capacitance, and inductance (RC extraction) for circuit simulation. In power- and thermal-optimization flows [4], net trace analyzes current density and IR drop in the power grid to avoid localized overheating or insufficient power supply.

At advanced technology nodes, the exponential growth in chip scale and complexity presents multiple challenges for layout net tracing, including soaring data volumes and increasingly intricate multi-layer connectivity relationships, thereby driving continuous innovation in EDA tools and algorithms.

Layout net tracing technology belongs to the field of physical verification for digital, analog, and RF layouts, and is mainly applied in the aforementioned stages of design verification, optimization, and manufacturing yield improvement. This contest simulates an industrial net trace scenario: the input is a target layout, its inter-layer connectivity relations, and the trace start point, and the output is the complete set of traced paths.

4. Competition Problem Description

Net tracing's most fundamental application takes as input the start point layer, the start point coordinate, and the inter layer connectivity rules, and then searches the layout file for the entire connected polygon region that contains the seed. The goal of this contest is to implement this function.

4.1 Input files

The contest provides one layout file plus three rule files.

1) Rule file: composed of a start point, Via rules, and a Gate rule. This contest uses a plain text file to describe the rule file, whose format is illustrated in Fig. 1. It contains two parts. The first part describes the start point information in the rule file. It begins with StartPos

(line 1). The line following specifies the start layer (L2) and its coordinate, indicating that net tracing begins at position (-65, 31) on layer L2. Line 3 marks the beginning of the second part, which describes the Via rules. It starts with Via; the next line (line 4) lists the Via relationship L1 L2 L3, meaning that L1 is connected to L3 through L2—that is, **polygons on L1 that intersect polygons on L2 are considered one connected region, and polygons on L2 that intersect polygons on L3 are likewise considered one connected region**.

```
1 StartPos
2 L2 (-65,31)
3 Via
4 L1 L2 L3
```

Figure 1a

```
1 StartPos
2 L4 (100,50)
3 Via
4 L1 L3 L4
5 L2 L3 L4
```

Figure 1b

Please note the following information:

a) Each rule file may contain one or two start points, one or more Via rules, and one Gate rule at most.

b) The coordinate values of a start point are 32-bit signed integers.

c) A rule file may include one or more Via rules. In every Via rule, each Layer name consists only of letters, digits, and underscores; each Via rule contains at least one Layer and must not repeat a Layer within the same rule. **For a Via rule that lists multiple Layers, polygons located on adjacent Layers that intersect each other are regarded as belonging to the same connected region, whereas polygons on non-adjacent Layers have no connectivity.** For example, in the single Via rule shown in Fig. 1a, polygons intersecting between L1 and L2 or between L2 and L3 are considered connected, but polygons intersecting between L1 and L3 are not. In the two Via rules shown in Fig. 1b, connectivity exists for the contact regions between L1 and L3, between L3 and L4, and between L2 and L3, yet there is no direct

connection between L1 and L4, between L2 and L4, or between L1 and L2. If duplicate connectivity relationships appear across multiple Via rules, the duplicates are ignored (e.g., the second definition of L3–L4 in Fig. 1b).

d) Each rule file may have one Gate rule or none. The Gate rule designates the Poly (polysilicon) layer and the AA (active-area) layer, exactly two layers in total—the first layer listed is Poly, and the second is AA. In the chip layout, if the Poly layer is tied to a logic-high signal, the AA regions on both sides of its intersection (the Source and Drain) are considered connected; conversely, if the Poly layer is tied to a logic-low signal, those AA regions are considered disconnected. As shown in Fig. 2a, when P1 is driven high and P2 is driven low, S1 and D1 are connected, whereas S2 and D2 are disconnected.

Polygon intersection criterion: Two polygons are deemed to intersect if they overlap in area with non-zero overlap, or their boundaries intersect—either at a point or along a line segment (**see the three cases in Fig. 2b: area overlap, boundary-point intersection, and boundary-segment intersection**).

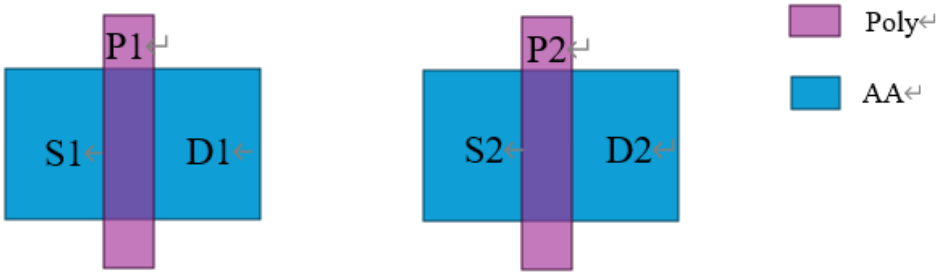

Figure 2a

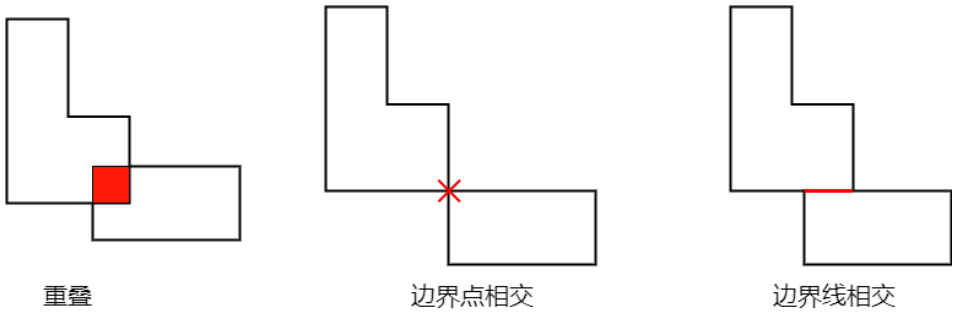

Figure 2b

2) Layout file:

a) Layout file: For this contest, the layout is given as a plain-text file rather than a standard GDSII or OASIS file. The file contains several layers, each consisting of one or more Manhattan polygons (hereinafter simply "polygons"). Polygons of the same layer are grouped under a Layer identifier (i.e., the layer name); the data for that layer continues until the next Layer identifier or the end of the file. Every line between two Layer identifiers lists, in counter-clockwise order, the coordinates of one polygon's vertices. The format, illustrated in Fig. 3, contains polygon data for each Layer. Line 1 shows the first Layer identifier, L1. Lines 2 – 3 list two polygons on L1. Line 4 shows the second Layer identifier, L2. Line 5 lists one polygon on L2. Line 6 shows the third Layer identifier, L3. Lines 7 – 9 list three polygons on L3. All polygons in Fig. 3 are shown in Fig. 4. Every shape is drawn inside a dashed-border rectangle; the legend in the rectangle's upper-right corner indicates the fill color for polygons on each Layer. An annotation of the form #Layer@#num marks a polygon as the num-th polygon on Layer (where num matches the order in the input file). For example, L1@1 denotes the first polygon on layer L1.

- All input polygons are guaranteed to be Manhattan (all edges parallel or perpendicular to the axes) and contain no holes or self-intersections.

- All polygon vertex coordinates are 32-bit signed integers.
- Layer identifiers contain only letters, digits, and underscores.

```
L1
(-5,-15),(10,-15),(10,10),(-5,10)
(15,-4),(5,-4),(5,-22),(23,-22),(23,5),(15,5)
L2
(0,0),(0,38),(-13,38),(-13,15),(-36,15),(-36,0)
L3
(-49,28),(-29,28),(-29,78),(-49,78)
(-73,14),(-11,14),(-11,40),(-73,40)
(-15,-25),(10,-25),(10,-17),(-15,-17)
```

Figure 3

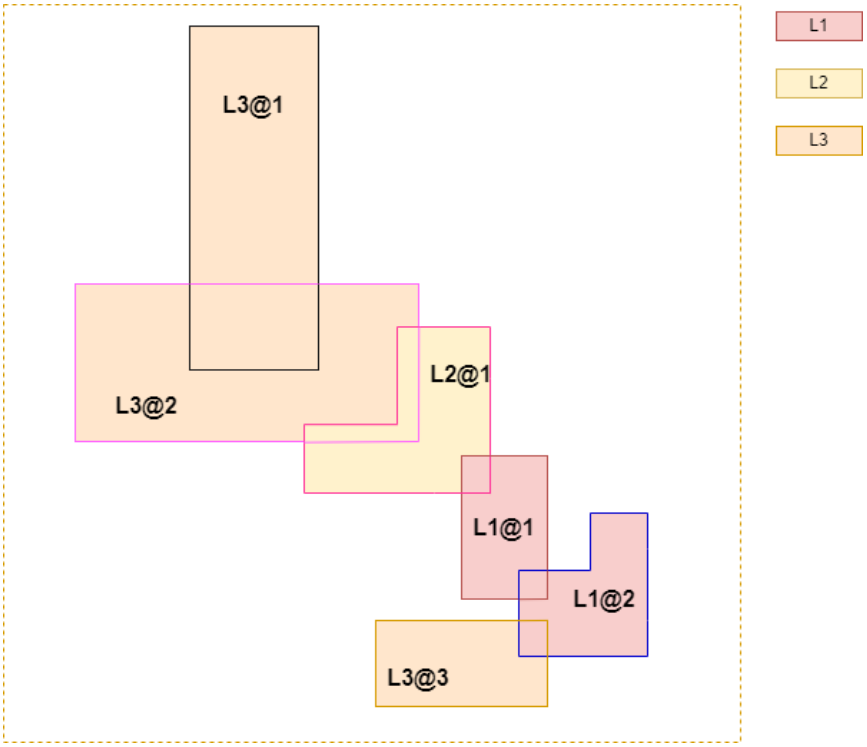

Figure 4

4.2 Net-trace requirements:

1) Starting from the start point given in the rule file, search the layout file for polygons that are connected to the start-point while satisfying the Via and Gate rules. Output the coordinates of all polygons reachable from the start-point that satisfy those rules. The output polygons must be grouped by layer, and the vertices of every polygon must be listed in

counter-clockwise order. For example, Figure 5 shows the result of tracing with the Via rule "L1 L2 L3" in the layout of Figure 3. The red cross marks the start point defined in the rule file, which lies inside the second polygon on layer L3. The polygons outlined in bold red are the ones found to be connected to the seed. The corresponding result file should therefore look like Figure 6. Specifically, the polygons returned for each layer are: L1 @ 1, L1 @ 2 on layer L1; L2 @ 2 on layer L2; and L3 @ 1, L3 @ 2 on layer L3. Because layers L3 and L1 are not adjacent in the Via rule (L1 ↔ L3 is not specified), L3 @ 3 cannot be reached via L1 @ 2.

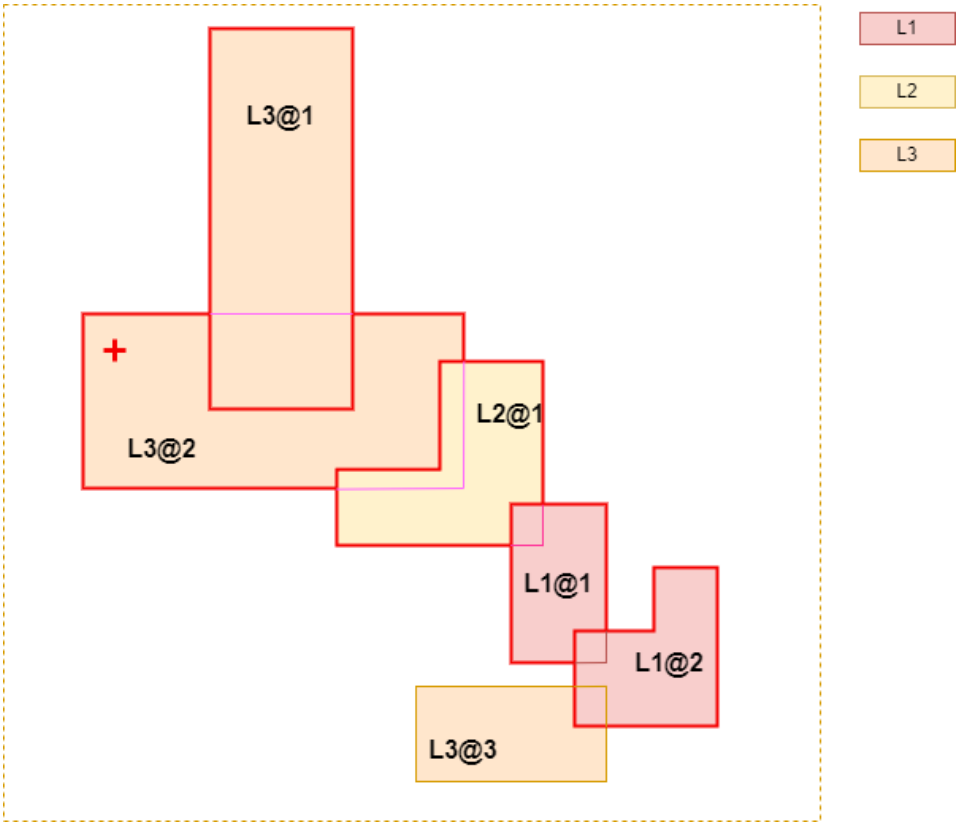

Figure 5

```
1 L1
2 (-5,-15),(10,-15),(10,10),(-5,10)
3 (5,-22),(23,-22),(23,5),(15,5),(15,-4),(5,-4)
4 L2
5 (0,0),(0,38),(-13,38),(-13,15),(-36,15),(-36,0)
6 L3
7 (-49,28),(-29,28),(-29,78),(-49,78)
8 (-73,14),(-11,14),(-11,40),(-73,40)
```

Figure 6

2) This contest comprises three sub-tasks:

The first sub-task tests net tracing with a single layer and a single start point. There is only one via rule and one start position, and the via rule contains only that one layer—the same layer on which the start position lies. In the layout shown in Figure 7, the start point is marked by a red cross. Starting from this start point, the correct connected-polygon set should include all polygons in the layout.

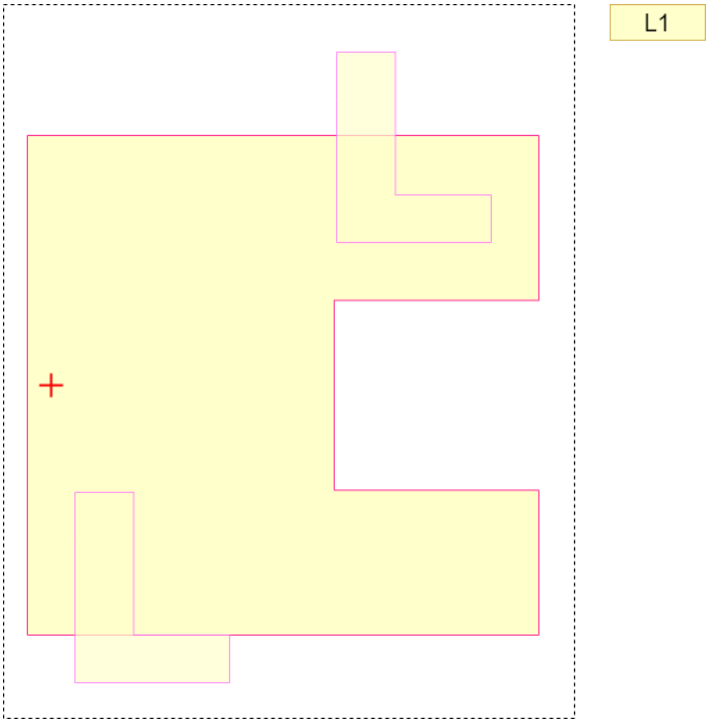

Figure 7

The second sub-task examines net tracing across multiple layers with a single start point. Using the layout file of Figure 3 and the rule file of Figure 1, a contestant must consider the Via rules to find connected polygons that span different layers. The output must list every polygon, on every layer, that is reachable from the start position.

The third sub-task emulates net tracing while accounting for the on/off states of transistors in the chip's logic circuitry. Here, the first start point s1 is taken to be the location where an

electrical signal is applied to the Poly layer. Once a logic-high signal is injected at s1, every Poly that is electrically connected to s1(according to the multi-layer Via rules) is driven high, whereas all other Poly are considered low. Under this condition, begin a new search from the second start point s2 and, again taking all Via rules into account, output all polygons on all layers that are connected to s2. **Polygons connected to s1 do not need to be included in the output.**

If the rule file in Figure 9 is applied to the layout file of Figure 8, the resulting trace is the output file shown on the righthand side of Figure 9; its tracing result is given in Figure 10. In Figure 10, polygons outlined in black belong to the connected region of the first start point, whereas those outlined in red belong to the connected region of the second start point. The black cross marks the first start position, and the red cross marks the second.

In this example a Gate rule is present: PO designates the Poly layer and AA designates the AA layer. According to the Via rule, M1 is connected to CT, and CT is connected to PO. Therefore, starting from the first start point s1 on the M1 layer at (118 743, 438 448), the polygon on the left-hand side of the PO layer in Figure 10 is driven to a logic high, and the AA polygon above it becomes electrically continuous on both sides of that Poly. Conversely, the Poly polygon on the right side is not connected to s1 and hence remains at logic low; the AA regions that it crosses are therefore disconnected. Consequently, the polygons outlined in red in Figure 10 constitute the complete set reachable from s2, and their coordinates are exactly the output required by the task. Please note:

- In the input layout, if an AA-layer polygon P1 intersects a Poly-layer polygon P2, then P2 always crosses P1 completely, i.e. P2 splits P1 into two parts.

- If an AA polygon P1 is connected (via the Via rules) to seed s2 and intersects a Poly polygon P2, then compute the polygons obtained by cutting P1 with P2, and then include these sliced polygons in the final result. For example, the upper AA polygon in Figure 10 is split by the crossing Poly into AA @ 1, AA @ 2, and AA @ 3. Because the Poly segment that crosses AA @ 1 and AA @ 2 is connected to s1 and therefore at logic high, AA @ 1 and AA @ 2 are connected. By the same way, the lower AA polygon is cut into AA @ 4, AA @ 5, and AA @ 6. However, the Poly segment that cuts AA @ 5 and AA @ 6 is not connected to s1, so AA @ 5 is not included in the trace results from s2.

```
AA
(118870,436985),(119770,436985),(119770,437985),(118870,437985)
(118820,439200),(119720,439200),(119720,440200),(118820,440200)
PO
(119460,438155),(119460,436865),(119560,436865),(119560,438385),
(119510,438385),(119510,440310),(119410,440310),(119410,438155)
(119080,438145),(119080,436865),(119180,436865),(119180,438375),
(119130,438375),(119130,440310),(119030,440310),(119030,438375),
(118890,438375),(118890,438145)
CT
(119580,440050),(119710,440050),(119710,440180),(119580,440180)
(119200,440050),(119330,440050),(119330,440180),(119200,440180)
(118830,440050),(118960,440050),(118960,440180),(118830,440180)
(119580,439770),(119710,439770),(119710,439900),(119580,439900)
(119200,439770),(119330,439770),(119330,439900),(119200,439900)
(118830,439770),(118960,439770),(118960,439900),(118830,439900)
(119580,439490),(119710,439490),(119710,439620),(119580,439620)
(119200,439490),(119330,439490),(119330,439620),(119200,439620)
(118830,439490),(118960,439490),(118960,439620),(118830,439620)
(119580,439210),(119710,439210),(119710,439340),(119580,439340)
(119200,439210),(119330,439210),(119330,439340),(119200,439340)
(118830,439210),(118960,439210),(118960,439340),(118830,439340)
(119420,438245),(119550,438245),(119550,438375),(119420,438375)
(118900,438235),(119030,438235),(119030,438365),(118900,438365)
(119630,437845),(119760,437845),(119760,437975),(119630,437975)
(118880,437835),(119010,437835),(119010,437965),(118880,437965)
(119630,437565),(119760,437565),(119760,437695),(119630,437695)
(118880,437555),(119010,437555),(119010,437685),(118880,437685)
(119630,437285),(119760,437285),(119760,437415),(119630,437415)
(118880,437275),(119010,437275),(119010,437405),(118880,437405)
(119630,437005),(119760,437005),(119760,437135),(119630,437135)
(118880,436995),(119010,436995),(119010,437125),(118880,437125)
M1
(119580,438715),(119710,438715),(119710,441225),(119580,441225)
(118830,438715),(118960,438715),(118960,441225),(118830,441225)
(118650,438165),(119030,438165),(119030,438585),(118650,438585)
(118880,436813),(119010,436813),(119010,438035),(118880,438035)
(119420,438175),(119765,438175),(119765,438585),(119420,438585)
(119160,438715),(119160,436935),(119760,436935),(119760,437885),
(120020,437885),(120020,437815),(120150,437815),(120150,438085),
(120020,438085),(120020,438015),(119760,438015),(119760,438045),
(119630,438045),(119630,437065),(119290,437065),(119290,438715),
(119400,438715),(119400,440180),(119130,440180),(119130,438715)
```

Figure 8

```
1 StartPos
2 M1 (118743,438448)
3 M1 (118871,441132)
4 Via
5 AA CT M1
6 PO CT M1
7 Gate
8 PO AA
```

```
1 AA
2 (118820,439200),(119030,439200),(119030,440200),(118820,440200)
3 (119130,439200),(119410,439200),(119410,440200),(119130,440200)
4 (119560,436985),(119770,436985),(119770,437985),(119560,437985)
5 CT
6 (118830,440050),(118960,440050),(118960,440180),(118830,440180)
7 (118830,439770),(118960,439770),(118960,439900),(118830,439900)
8 (118830,439490),(118960,439490),(118960,439620),(118830,439620)
9 (118830,439210),(118960,439210),(118960,439340),(118830,439340)
10 (119200,440050),(119330,440050),(119330,440180),(119200,440180)
11 (119200,439770),(119330,439770),(119330,439900),(119200,439900)
12 (119200,439490),(119330,439490),(119330,439620),(119200,439620)
13 (119200,439210),(119330,439210),(119330,439340),(119200,439340)
14 (119630,437845),(119760,437845),(119760,437975),(119630,437975)
15 (119630,437565),(119760,437565),(119760,437695),(119630,437695)
16 (119630,437285),(119760,437285),(119760,437415),(119630,437415)
17 (119630,437005),(119760,437005),(119760,437135),(119630,437135)
18 M1
19 (118830,438715),(118960,438715),(118960,441225),(118830,441225)
20 (119160,438715),(119160,436935),(119760,436935),(119760,437885),
   (120020,437885),(120020,437815),(120150,437815),(120150,438085),
   (120020,438085),(120020,438015),(119760,438015),(119760,438045),
   (119630,438045),(119630,437065),(119290,437065),(119290,438715),
   (119400,438715),(119400,440180),(119130,440180),(119130,438715)
```

Figure 9 The left figure is the rule file, and the right figure is the result file.

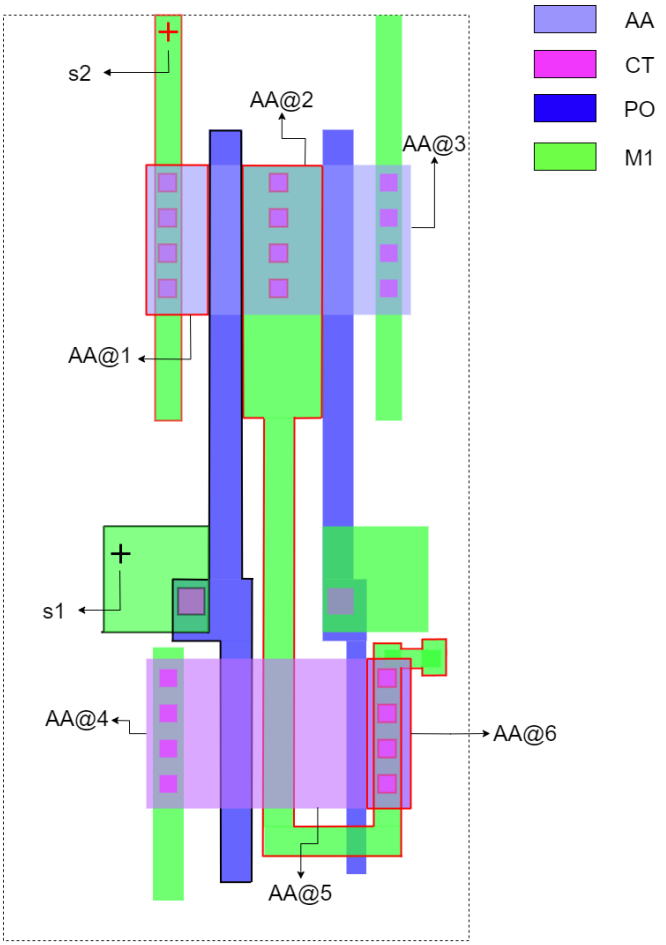

Figure 10

4.3 Output file format

The output file must be a Linux-based text file. It shall contain the complete union of

polygons discovered in the current layout for all start points defined in the rule file, grouped by layer and with one polygon per line. The exact notation for each polygon follows the specification given in the Net-trace Requirements section.

4.4 Running syntax

trace -layout ./layout.txt -rule ./rule.txt [-thread n] -output ./res.txt

trace: an executable Net Trace program, compiled by each contestant on the contest server.

Parameter settings:

-layout: required. Path to the layout file provided by the organizer.

-rule: required. Path to the rule file provided by the organizer.

-thread: optional. Specify n threads for parallel computation.

-output: required. Path to the result file to be generated by the program.

4.5 Submission

All participating teams must submit the following files after finishing the task:

1) The compiled executable program and any required runtime libraries;

2) Information regarding all third-party libraries used during development, including the libraries names and their official websites.

3) Source code is encouraged but not mandatory.

5. Scoring Criteria

The scoring for this task is divided into three parts: Correctness, Performance, and Multi-threaded parallelism. The criteria for each part are as follows:

5.1 Correctness:

60 points in total. The task contains three sub-tasks; each sub-task that is completely correct,

with no erroneous or missing polygons, earns 20 points. For every erroneous or missing polygon, 1 point is deducted until the score reaches zero. The sum of the three sub-problem scores is the total correctness score.

5.2 Performance:

40 points in total. Split into single-thread and multi-thread portions, each worth 20 points. Performance points are awarded only if the correctness score is full.

The single-thread performance score is determined by ranking Trace's **single-process, single-thread** runtime. For all contestants who achieved a perfect correctness score, runtimes are ordered from smallest to largest; 1st place gets 20 points, and each subsequent rank loses 20 / (number of ranked contestants − 1) points. **If the program uses multiple processes or threads when the -thread option is not set, this score is 0 points.**

The multi-thread performance score is determined by ranking Trace's single-process, 8-thread runtime (the test environment guarantees at least 8 CPU cores). If multi-threading is not implemented, the single-thread runtime is used. For all contestants who achieved a perfect correctness score, runtimes are ordered from smallest to largest; 1st place gets 20 points, and each subsequent rank loses 20 / (number of ranked contestants − 1) points. If the program's maximum thread count exceeds the number specified by **the -thread option, this score is 0 points.**

5.3 Additional information:

If, on the hidden test case, the single-process single-thread runtime exceeds 8 hours, the submission receives 0 points for both correctness and performance.

6. References

1. Pandey R, Agrawal N, Arghavani R, et al. Analysis of local interconnect resistance at scaled process nodes[C]//2015 73rd Annual Device Research Conference (DRC). IEEE, 2015.DOI:10.1109/DRC.2015.7175620.

2. 余建波, 卢笑蕾, 宗卫周.基于局部与非局部线性判别分析和高斯混合模型动态集成的晶圆表面缺陷探测与识别[J]. 自动化学报, 2016, 42(1):13.DOI:CNKI:SUN:MOTO.0.2016-01-005.

3. Chen D C, Lin G S, Lee T H, et al. Compact modeling solution of layout dependent effect for FinFET technology[C]//2015 International Conference on Microelectronic Test Structures (ICMTS).IEEE, 2015.DOI:10.1109/ICMTS.2015.7106119.

4. Pandey R, Agrawal N, Arghavani R, et al. Analysis of local interconnect resistance at scaled process nodes[C]//2015 73rd Annual Device Research Conference (DRC). IEEE, 2015.DOI:10.1109/DRC.2015.7175620.

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