

# 器件重叠检查方法的研究

薛长青<sup>1</sup> 李彦祯<sup>1</sup> 于洪真<sup>1</sup> 李发春<sup>1</sup>

XUE Changqing LI Yanzhen YU Hongzhen LI Fachun

## 摘要

芯片版图设计的准确性和可靠性对芯片性能至关重要，其中器件重叠检查是核心环节。为全面检查器件重叠情况，文章提出了一种器件重叠规则检查方法，统一定义版图数据结构，分类处理多边形和复合多边形器件，将器件重叠情况细分为复合多边形与复合多边形器件重叠、多边形（不包括矩形）与多边形重叠、多边形（不包括矩形）与复合多边形重叠、矩形多边形与矩形多边形重叠、矩形多边形与复合多边形重叠、矩形多边形与多边形（不包括矩形）重叠以及自定义器件（合法重叠）七种情况，同时运用多种重叠判断算法。结果显示，该方法能够有效检测出器件的重叠，显著提高了版图设计的准确性和效率。

## 关键词

芯片；多边形器件；复合多边形器件；器件重叠；自定义器件

doi: 10.3969/j.issn.1672-9528.2025.03.044

## 0 引言

芯片作为计算的核心部件，其版图设计至关重要。传统版图设计工具在处理器件重叠检测时，主要依赖于遍历所有器件并进行相互比较的方法。然而，随着科技的发展，器件种类不断增多，包括基本器件、器件以及各类自定义器件，而且其版图设计数据结构也与以前版图设计存在显著差异。当前版图设计工具尚处于初步发展阶段，不同工具间对应的版图设计数据结构尚未统一，这进一步加剧了器件重叠检测的复杂性。

针对这一现状，本文提出了一种器件重叠规则检查方法。该方法首先定义了版图设计工具的数据结构，该结构涵盖了器件类型、器件名称、器件坐标以及器件参数等关键信息。其中，器件类型被细分为多边形器件和复合多边形器件两大类，分别对应一维数组和多维嵌套数组的坐标表示方式。在多边形类型器件中，sym1 作为典型代表，而在复合多边形器件中，sym2、sym3 等则占据了重要地位。此外，该方法还充分考虑了自定义器件的特殊性，如自定义 sym5 器件等，这些器件同样可以根据其形状特征被归类为多边形器件或复合多边形器件<sup>[1-3]</sup>。

在器件重叠检测的具体实践中，本文方法将器件重叠情况细分为复合多边形与复合多边形器件重叠、多边形（不包括矩形）与多边形重叠、多边形（不包括矩形）与复合多边形重叠、矩形多边形与矩形多边形重叠、矩形多边形

与复合多边形重叠、矩形多边形与多边形（不包括矩形）重叠以及自定义 sym5（合法重叠）七种情况。针对每种情况，本文方法都定义了相应的器件重叠数据结构，包括器件重叠输出信息、器件重叠对象名称以及器件重叠对象坐标等关键信息。

## 1 版图数据结构定义

### 1.1 器件类型与坐标表示

在器件重叠规则检查的方法中，器件类型与坐标表示构成了版图数据结构的核心要素。该方法针对版图设计工具所对应的特定数据结构进行了设计，以确保能够准确、高效地检测器件之间的重叠情况。

首先，在器件类型方面，该方法将器件划分为多边形器件和复合多边形器件两大类。多边形器件以其简单的几何形态和易于处理的坐标表示方式，成为版图设计中的基础元素。其中，sym1 器件作为多边形器件的典型代表，广泛应用于各种电路的连接与封装。而复合多边形器件以复杂多变的形状，为版图设计提供了丰富的选择。sym2 器件、sym3 器件等均属于复合多边形器件的范畴，在计算与通信等领域发挥着至关重要的作用。

此外，该方法还充分考虑到自定义器件的特殊性，如自定义 sym5 器件等，虽然形态各异，但同样可以根据其形状特征被归类为多边形或复合多边形。为区分自定义器件与普通多边形器件，该方法在器件类型中引入了自定义类型标识。当自定义类型标识为空时，该器件即被视为普通多边形器件；而当其被赋予特定值时，则表明该器件为自定义器件，需要按照特定的规则进行处理。

1. 山东云海国创云计算装备产业创新中心有限公司  
山东济南 250101

在坐标表示方面，该方法为多边形器件和复合多边形器件分别定义了不同的坐标表示方式。对于多边形器件，其坐标采用一维数组的形式进行表示，每个坐标点由横纵坐标组成，这些坐标点共同构成了器件的几何形状。而对于复合多边形器件，由于其形状更为复杂，因此采用多维嵌套数组的形式来表示其坐标。

### 1.2 自定义器件处理

自定义器件也是多边形器件的一种，但具有特殊的自定义类型标识。自定义 sym5 由多个矩形组合而成，通过自定义类型标识可以区分多边形器件和自定义矩形多边形器件。

## 2 重叠规则检查方法

### 2.1 器件模型定义与解析

根据版图数据结构，对器件进行模型定义和解析。多边形器件模型包含名称、类型、自定义器件类型、器件坐标和器件参数等信息。复合多边形器件模型则包含类似的信息，但器件坐标为多维嵌套数组。

### 2.2 关键技术实现

#### 2.2.1 多边形器件（不包括矩形）与多边形器件（不包括矩形）重叠

解析版图设计数据，得到了所有器件模型信息的数组。随后，根据器件的类型进行了分类保存：对于非自定义的多边形器件，将其坐标集合保存在 listPoly 中。接下来，进行多边形器件（排除自定义器件）的重叠检查。获取每个多边形器件的坐标信息作为器件的唯一标识，并应用多边形重叠规则进行判断。这一规则基于射线法，即判断一个多边形器件的每个坐标点是否位于另一个多边形器件的内部。若存在至少一个坐标点满足此条件，则确认两个器件重叠。对于重叠的器件，构建了相应的数据结构进行封装，并提供了页面异常展示功能，以多边形器件 sym1 为例，实验效果如图 1 所示。



图 1 sym1 器件重叠图

#### 2.2.2 多边形器件（不包括矩形）和复合多边形器件重叠

解析版图设计数据，得到了所有器件模型信息的数组。随后，根据器件的类型进行了分类保存：对于非自定义的多边形器件，将其坐标集合保存在 listPoly 中；而对于复合多边形器件，由于其结构的复杂性，将其坐标集合以嵌套列表的形式保存在 listGroups 中。接下来，进行多边形器件（不包括矩形）和复合多边形的重叠检查。对于每个器件，获取其坐标信息以唯一标识。应用多边形器件与复合多边形器件的重叠规则来判断是否存在重叠情况。具体而言，该规则涉及分别获取多边形器件和复合多边形器件（后者包含多个简单多边形）的坐标点，并检查多边形器件是否与复合多边形器件中的每一个简单多边形都发生重叠。如果多边形器件与复合多边形器件中的至少一个简单多边形重叠，则认定多边形器件与复合多边形器件重叠，并构建相应的数据结构来封装输出所有重叠的器件信息，同时在页面上提供异常展示。若多边形器件与复合多边形器件中包含的所有简单多边形均未发生重叠，则判定两者不重叠。

#### 2.2.3 复合多边形器件和复合多边形器件重叠

解析版图设计数据，得到了所有器件模型信息的数组。随后，根据器件的类型进行了分类保存：对于复合多边形器件，由于其结构的复杂性，将其坐标集合以嵌套列表的形式保存在 listGroups 中。随后，按照器件分类得到所有组合多边形器件集合，并获取每个器件的复合多边形坐标以标识器件。接下来，通过复合多边形器件和复合多边形器件的重叠规则，来确认是否存在器件重叠。具体的重叠判断规则步骤<sup>[4-6]</sup>如下。

##### (1) 获取坐标点

分别获取两个待比较的复合多边形器件的所有多边形坐标点。

##### (2) 重叠判断

对于一方复合多边形中包含的所有多边形，依次检查它们是否与另一方复合多边形中包含的所有多边形重叠。这里所说的“重叠”，是指两个多边形相交。

##### (3) 构建输出

如果在一方复合多边形中找到了至少一个多边形与另一方复合多边形中的某个多边形重叠，则判定这两个复合多边形重叠。此时，将构建封装输出所有重叠的器件结构数据，并提供页面异常展示，以便用户直观地看到哪些器件发生了重叠。

##### (4) 无重叠情况

如果在一方复合多边形中找不到任何与另一方复合多边形中的多边形重叠的情况，则判定这两个复合多边形不重叠。

### 2.2.4 矩形器件和复合多边形器件重叠

首先，通过数据获取与解析机制，将版图设计数据序列化，并转化为一个包含器件模型信息的对象数组。

接着，根据器件类型的差异，进行了细致的分类存储。对于结构相对简单的自定义多边形矩形器件，其对象坐标集合被保存在 `listRectof` 列表中。而对于结构更为复杂的复合多边形器件，则采用了嵌套列表的形式，保存于对象集合 `list-Groups` 列表中。

随后，进行矩形多边形与复合多边形的重叠检查。为每个矩形多边形器件提取了坐标信息，并将其作为器件的唯一身份标识。然后，应用矩形多边形与复合多边形重叠判断的规则。这一规则的核心在于，判断矩形器件所占据的最大区域范围是否与复合多边形中的任意一个多边形存在交集。一旦确认矩形器件与复合多边形中的某个多边形相交，即判定这两个器件发生了重叠。

对于检测出的重叠器件，把重叠器件的数据结构进行了封装，提供了直观的页面异常展示功能。

### 2.2.5 矩形器件和矩形器件重叠，排除 sym5 器件

在器件版图设计的重叠规则检查流程中，首先需要从设计数据中提取并解析出自定义的多边形矩形器件对象集合。这一步骤涉及数据的获取、序列化处理以及基于器件类型的筛选。随后，识别出符合 `sym5` 定义的矩形器件组合。`sym5` 特指由 3 个矩形构成的结构，其中 1 个矩形与 2 个矩形相交<sup>[7-10]</sup>。

为了找到这些 `sym5` 组合，遍历矩形器件对象集合，并对每个矩形器件执行重叠判断。在此过程中，利用了一个辅助函数 `checkRectAndRect`，该函数接收 3 个参数：器件组合的列表、多边形对象的列表以及矩形器件对象的列表。该函数的核心任务是识别并标记出需要删除的 `sym5` 矩形器件，这些器件的特征就是 3 个自定义矩形器件，且 1 个矩形器件和 2 个矩形器件合法重叠。

在 `checkRectAndRect` 函数的内部实现中，首先初始化了一个用于存储消息的列表和一个用于记录待删除矩形索引的列表。接着，遍历矩形器件列表，为每个矩形创建一个 `Rectangle2` 对象，并检查是否与其他矩形重叠。如果当前矩形与至少一个其他矩形重叠，将这些重叠矩形的索引添加到一个临时列表中。

随后，根据重叠情况判断当前矩形是否为 `sym5` 的一部分。如果是，将当前矩形的索引也添加到临时列表中，并将整个列表的内容合并到待删除的索引列表中。

为避免在删除元素时产生索引冲突，对待删除索引列表进行降序排序，并从后往前遍历该列表，根据索引从矩形器件列表中删除相应的矩形。

最终，经过 `checkRectAndRect` 函数的处理，矩形器件列表将只包含那些不构成 `sym5` 的矩形器件。

当已经得到了不包含 `sym5` 的矩形器件对象集合后，接下来需要判断这些矩形器件之间是否存在重叠。为了实现这一点，可以采用基于范围区域获取比对的方法，即取每个矩形器件中所有坐标的最大值和最小值，是否有交叉区域，可以判断对应的矩形器件是否重叠。实验效果如图 2 所示。

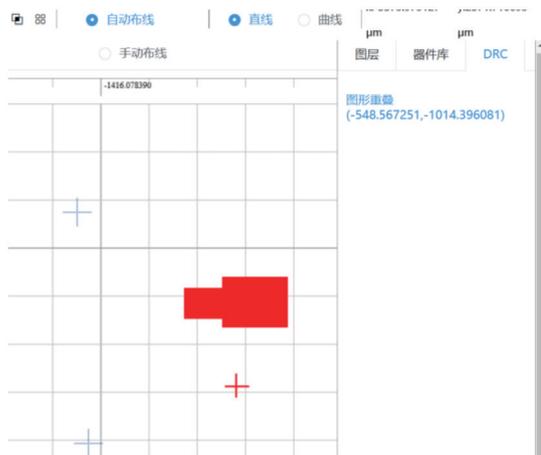


图 2 矩形器件和矩形器件重叠图

### 2.2.6 矩形器件和多边形器件（不包括矩形）重叠

首先通过序列化解析版图数据获取矩形器件对象集合与不包括矩形器件的多边形器件对象集合。矩形器件和多边形器件（不包括矩形）重叠规则判断可以根据多边形器件和多边形器件规则判断。以矩形器件和 `sym1` 器件重叠为例，实验效果如图 3 所示。

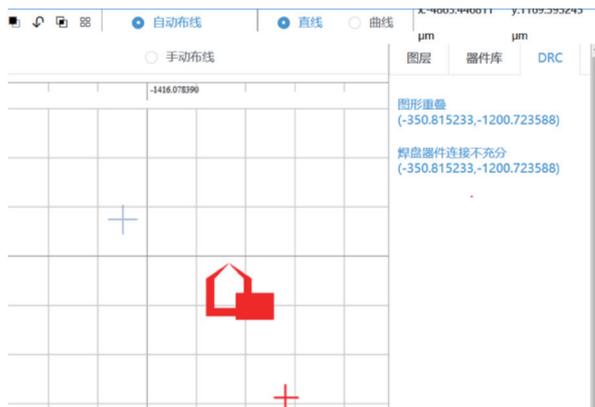


图 3 矩形器件和 `sym1` 器件重叠图

### 2.2.7 `sym5` 器件合法重叠

在器件重叠规则检查的过程中，自定义 `sym5` 的合法重叠情况比较特殊。首先，从自定义多边形器件集合中筛选出矩形器件，并利用索引算法高效地遍历这些器件组合，以找出存在重叠的器件。

具体而言，初始化一个空的索引集合，并对于每个矩形器件，再次遍历整个集合（排除自身），检查是否存在与其

重叠的其他矩形器件。若某个矩形器件与至少两个其他矩形器件重叠，初步判断这可能构成了一个自定义 sym5，即由一个桥面（当前矩形器件）和两个桥墩（重叠的矩形器件）组成。相关代码如下：

```
Set<Integer> rectIndices = new HashSet<>();
// 遍历所有矩形，查找重叠的矩形
for (int i = 0; i < listRectof .size(); i++) {
    List<DataPoint> rectangle = listRectof .get(i);
    Rectangle2 nowRectangle = new Rectangle2(rectangle);
    rectIndices.clear(); // 重置重合索引集合
    for (int j = 0; j < listRectof .size(); j++) {
        if (i != j) { // 不与自己比较
            List<DataPoint> othertangle = listRectof .get(j);
            Rectangle2 otherRectangle = new Rectangle2(othertangle);
            if (nowRectangle.overlaps(otherRectangle)) {
                rectIndices.add(j); // 添加重合矩形的索引
            }
        }
    }
    if (rectIndices.size() >= 2) {
        // 说明有两个矩形和这个矩形重叠，即 3 个矩形重叠，
        进一步判断是否符合 sym5
    }
}
```

其中，overlaps 为 2 个矩形重叠的方法。接下来，进一步验证 sym5 是否符合特定的规则。具体而言，定义了两个数组 minDistances1To2 和 minDistances1To3，分别用于存储桥面到第一个桥墩和第二个桥墩的最小水平和垂直距离。初始时，这两个数组的所有元素都被设置为正无穷大。接下来，遍历桥面上的所有点，对于每个点，计算它到第一个桥墩的最小水平和垂直距离，并更新 minDistances1To2 数组中的相应元素。同样地，再次遍历桥面上的所有点，计算它们到第二个桥墩的最小距离，并更新 minDistances1To3 数组。最后，比较 minDistances1To2 和 minDistances1To3 数组中对应元素的值。如果桥面到两个桥墩的最小水平和垂直距离都相等，即 (minDistances1To2[0] == minDistances1To3[0]) && (minDistances1To2[1] == minDistances1To3[1])，则认为 sym5 的分布是均匀的，符合自定义 sym5 的合法重叠规则。否则，系统会提示 sym5 分布不均匀，需要设计者进行相应的调整。

自定义 sym5 器件合法重叠，两端桥墩与中间线距离不相等，桥面与桥墩基础距离分布不均匀，实验效果如图 4 所示。

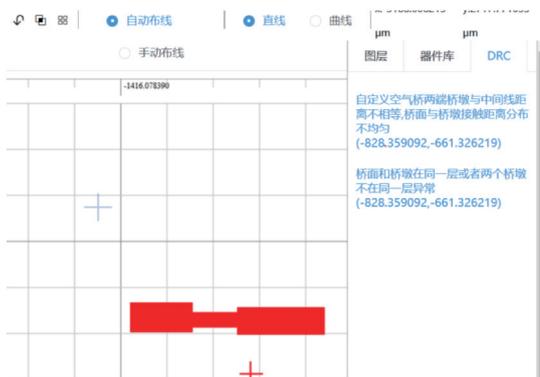


图 4 自定义 sym5 器件合法重叠，不相等不均匀图

### 3 结论

本文提出了一种器件重叠规则检查方法，通过定义版图数据结构、分类处理多边形和复合多边形器件以及采用多边形重叠判断算法实现了对器件重叠情况的全面检查。实验结果表明该方法具有显著的有效性和实用性。

#### 参考文献：

- [1] 杨素行. 模拟电子技术基础简明教程 [M]. 3 版. 北京: 高等教育出版社, 2006.
- [2] 王利丹. 提高可编程增益放大器准确度的一种简单方法 [J]. 北京电子科技学院学报, 2008(2):29-31.
- [3] 郭光明, 王秀华. 模拟数据采集系统误差的自动纠正 [J]. 电子与自动化, 2000(5):11-13.
- [4] 郑君里, 应启珩, 杨为理. 信号与系统:上册 [M].3 版. 北京: 高等教育出版社, 2011.
- [5] 张晓威, 苏淑靖. 基于 FPGA 的高速高精度数据采集系统的设计 [J]. 仪表技术与传感器, 2016(1): 70-72.
- [6] 陈兴尧. 备品备件管理系统在变电站二次缺陷管理中的应用 [J]. 机电信息, 2018(12): 49-50.
- [7] 王朝凯, 范永璞, 印海洋, 等. 一种基于 Git 的智能变电站二次回路信息模型的版本管理系统 [J]. 电气技术与经济, 2019(6): 41-43.
- [8] 林利祥, 陈创, 王幸, 等. 基于物联网 RFID 和 IC 卡技术的变电站图纸管理系统研究 [J]. 电工电气, 2020(5):61-65.
- [9] 周二彪, 李忠政, 赵军, 等. 基于三维引擎的变电站二次系统三维模型库设计及管理系统 [J]. 电气技术, 2020, 21(8): 58-65.
- [10] 许磊, 郑超, 孟洋洋, 等. 智能变电站二次装置一键式隔离系统研究 [J]. 电测与仪表, 2024,61(6):141-147.

#### 【作者简介】

薛长青(1983—), 男, 山东济南人, 本科, 中级工程师, 研究方向: 云平台、EDA 平台、物联网平台相关工程技术研究、开发。

(收稿日期: 2024-10-08)