2023 广东工业大学 EDA 组 23 级二轮考核项目

项目名称：简化布局器的实现

张伟祺 3123009536

一、项目实验结果

在本次项目中，成功实现了一个基于回溯算法对输入 case 进行求解的布局优化系统。该方案使实例间总线长最小化,有效降低了布线复杂度,达到了优化布局的设计目标。经过实验和测试,系统在布局优化方面取得了良好的效果。实验结果表明,该系统可以在较短的时间内找到最优解。

二、主要技术路线及实现方式

1、 读取输入文件

读出关键实例及连网信息,包括 die 大小、单元数目、单元名称、线网数目、每个线网的单元等数据。将其封装为自定义的结构体表示,包括 instance（包括单元名称，x、y 坐标，以及记录是否放置的标准）和 net（包括线网名称，线网的单元数量，线网的单元）。

2、 记录连接关系

根据连接关系构建邻接矩阵，矩阵值表示实例间是否连接,用于后续计算新增线长。在后续布局单元 x 时，即可以查看 x 与哪个单元连接并且该单元是否已经放置，然后计算布局单元 x 后所增加的线长。

3、 递归回溯搜索最优解：

递归函数 enumerate()进行布局优化搜索。函数通过遍历所有实例在 die 上的所有可能位置,通过 calculate_add_length（）函数计算增加的线长。如当前布局线长已超过当前最优值,则直接返回,剪枝优化;如果找到更优解,更新最优布局。该递归搜索实现了穷举探索,以得到全局最优布局。

4、结果输出

递归搜索得到的最终最优布局方案，依次输出单元的名称，x、y 坐标到输出到文件,完成结果展示。

三、遇到的主要问题及解决方法

问题 1：如何不遗漏的探寻所有的放置方案

解决方法：从表格的第一个位置开始，依照单元的输入顺序放置，放置完成一个单元后记录该位置已被放置，递归放置下一个单元。

问题 2：随单元数量规的增长,递归树枝数呈指数增长,计算时间随之剧增。

解决方法：是在递归过程中加入判断,当递归线长已超出当前最优值时剪枝返回,从而减少不必要的搜索,加速求解过程。

四、主要创新点

1、避免重复的运算

在布局新的单元后计算线长时，为避免重复计算，使用了 calculate_add_length(),只需要计算对比原来布局所改变了的单元所影响的线长，而不需要重新计算。

2、剪枝优化

在递归函数布局下一个单元时把以布局的单元的所有线网长度与所存储的目前最优布局方案的总线长进行对比，以为若当前总线长已经比最优解的总线长要更长了，那继续布局下一个单元只会增加线长，其线长必定大于目前最优解的总线长。因此，若当前线长比目前最优布局总线长要长，直接返回，完成剪枝，大大减少了计算量。

五、总结

该项目成功采用递归回溯算法求解了布局优化问题,基本上完成了项目目标，
算法核心在于递归搜索所有可能的布局,以找出最优解。优化手段是剪枝优化。
今后可研究引入模拟退火、遗传算法等方法查找布局方案。