

VLSI实验二报告_22336216+陶宇卓

算法逻辑与实现思路

1. 算法选择

本实验采用模拟退火算法（Simulated Annealing）作为核心优化方法，结合贪心策略的初始布局，以最小化总线长为目标进行FPGA布局设计。

2. 实现流程

1. 初始布局（`initializePlacement`）

- 模块选择：从未布局的模块中，选择与已布局模块连接密度最大的模块。连接密度通过模块所在网络中的已布局邻居数量计算。
- 位置选择：遍历所有空位，计算将模块放置到每个位置后的总线长（目标函数），选择代价最小的位置进行放置。
- 目标：通过贪心策略为后续模拟退火提供较优的初始解，减少退火阶段的搜索时间。

2. 模拟退火优化（`myplacementAlgorithm`）

- 参数设置：初始温度（`T=1000`）、降温速率（`alpha=0.98`）、最低温度（`T_min=1e-5`）。
- 扰动操作：
 - 交换操作：随机选择两个未固定模块，交换其位置，计算新总线长。
 - 移动操作：随机选择一个未固定模块，将其移动至随机空位。
- 接受准则：
 - 若新解更优（总线长减少），则接受；
 - 否则，以概率 $\exp(-\Delta/T)$ 接受较差解，避免陷入局部最优。
- 降温过程：每轮迭代后按 `T *= alpha` 降温，直至温度低于阈值。
- 最优解保存：记录退火过程中找到的最优解，最终恢复至全局状态。

3. 约束处理

- 固定模块：通过 `inst->isFixed()` 跳过已固定模块的操作。
- 位置冲突：移动或交换时检查目标位置是否为空，避免模块重叠。

- 合法性验证: `reportValid()` 函数检查实例位置与块的一致性, 确保布局合法。

4. 算法创新点

- 混合策略: 结合贪心初始布局与模拟退火, 提升收敛速度。
- 双扰动操作: 同时支持交换和移动操作, 增强搜索多样性。
- 动态接受概率: 通过指数函数控制较差解的接受概率, 平衡探索与开发。

实验结果

(以下数据需根据实际运行结果补充)

数据集	优化后总线长	运行时间 (s)
small	444	8
med1	3989	35
med2	4627	37
lg1	14983	129
lg2	18037	140
xl	37026	525
huge	76247	1660

实验总结

优点

1. 全局优化能力: 模拟退火有效跳出局部最优, 尤其在初始布局基础上进一步优化。
2. 灵活性: 支持多种扰动操作, 适应不同规模的FPGA布局问题。
3. 鲁棒性: 通过合法性检查确保解的有效性。

缺点

1. 计算开销大: 遍历空位和频繁评估总线长导致时间复杂度较高。

2. **随机性依赖**：解的质量受随机扰动操作影响，可能需多次运行取最优。
3. **初始布局效率**：贪心策略在大型FPGA上可能耗时过长。

改进方向

1. **加速评估**：增量式计算总线长，避免全量重新评估。
2. **启发式初始布局**：引入聚类分析或力导向布局，提升初始解质量。
3. **并行化**：利用多线程加速退火过程中的扰动操作。