

# 2024 中国研究生创“芯”大赛·EDA 精英挑战赛 赛题指南

版本	时间	修订内容
V0.1	2024-08-01	初版
V0.2	2024-08-26	更新线长分数计算方式
V0.3	2024-09-10	更新 FPGA 架构说明 增加 PLB 作为 DRAM 使用时的资源描述 增加 IPPIN 类型资源描述 每个 clock region 内时钟总数为 28 更新 Arch 文件格式描述

## 一、赛题名称：

FPGA 面向布线优化的详细布局

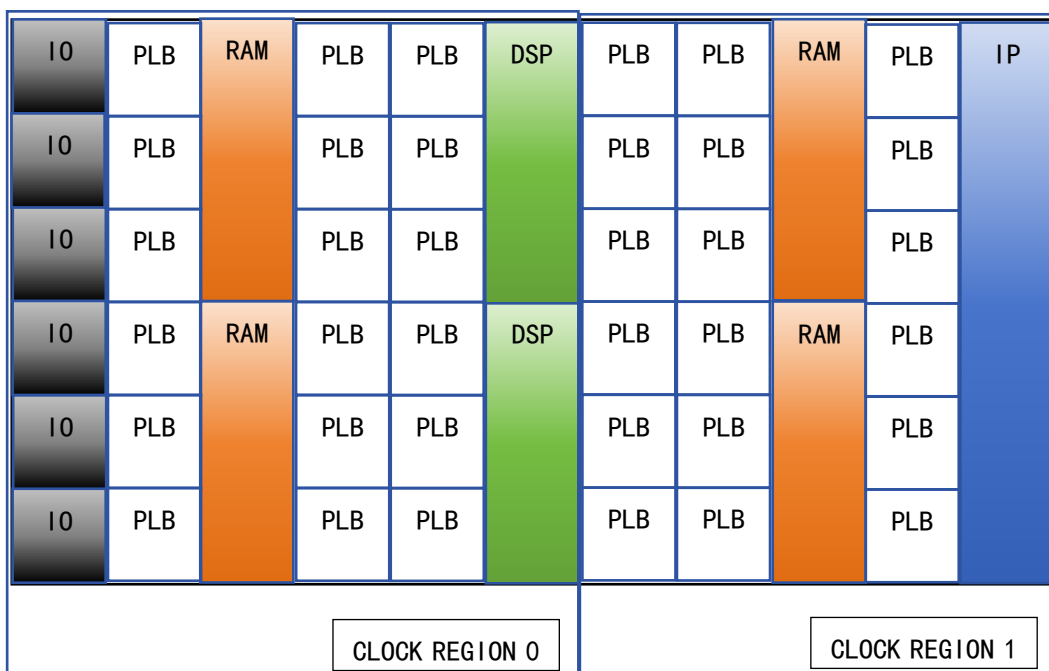
## 二、命题单位：

上海安路信息科技有限公司

## 三、赛题主席

贺旭（湖南大学）

## 四、赛题背景：



图一 FPGA 逻辑阵列示意图

### 1、FPGA 架构说明

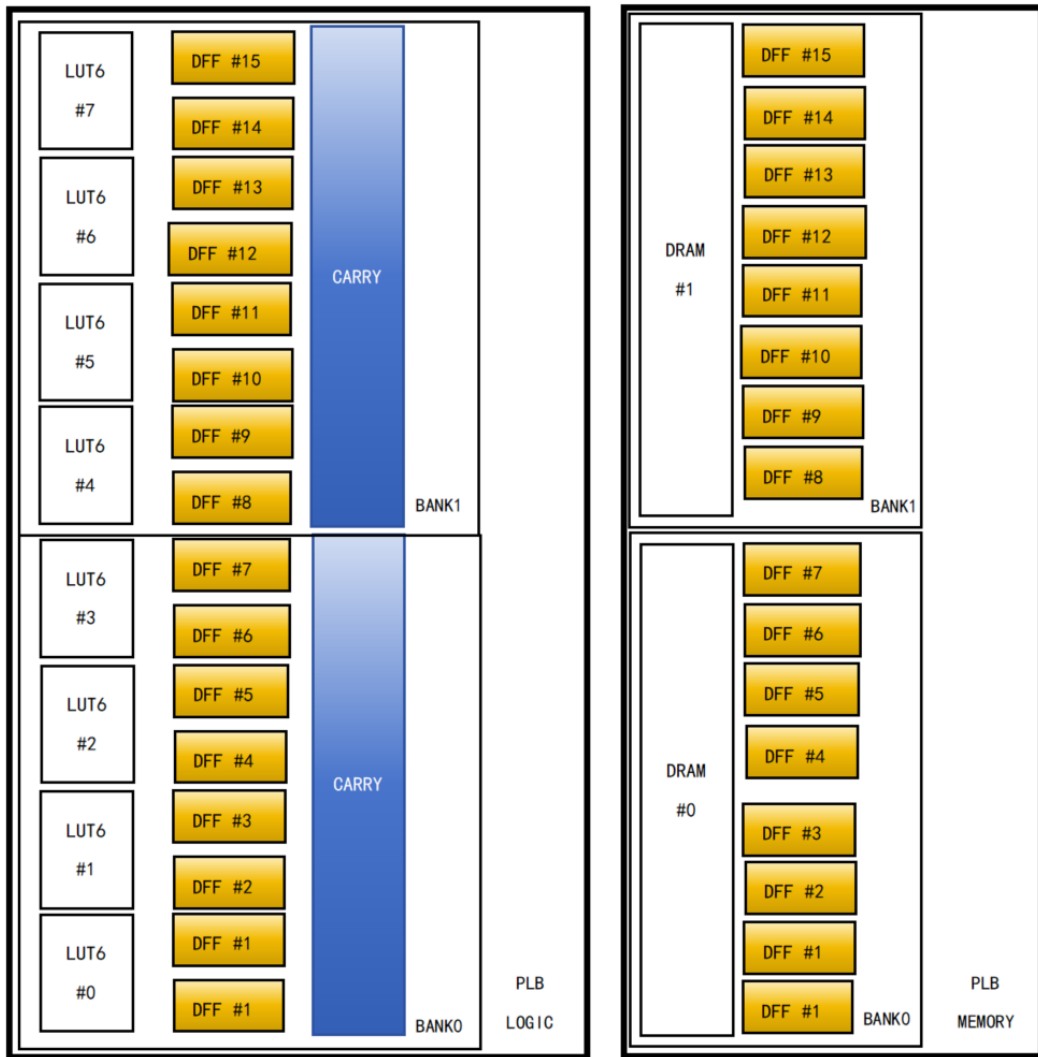
大规模可编程器件（FPGA）由多种逻辑资源所构成。包含可编程逻辑模块（PLB），块 RAM，DSP，IO 及复杂 IP 模块。各种资源在芯片内以列状形式排列成阵列，如图一所示。其中 **PLB** 作为最基础逻辑块，可以实现基本的组合逻辑查找表和时序逻辑功能，同时还可以提供专用的快速进位链，以执行快速算术加法和减法。RAM 及 DSP 资源与 PLB 资源等宽，占据整数个 PLB 资源高度。若干个 PLB 及专用逻辑单元共享相同的时钟资源，即相同的 Clock Region 如图所示。每个 clock region 内可以容纳一定数量限制的时钟资源（28 路时钟）。

本赛题中，如图二所示，假设每个 PLB 包含 2 个独立区域 (BANK)，每个 BANK 中包含 4 个 6 输入 2 输出查找表(LUT)，8 个触发器 (DFF) 及快速进位链。PLB 中 LUT 由下至上依次编号为 0~7，DFF 由下至上依次编号为 0~15。相同 bank 内的 8 个 DFF，要求时钟及 reset 信号必须相同，ce 信号不超过 2 个。另外每个 PLB 单元可以实现特定比特位的进位链，并且可以级联纵向相邻的 PLB 实现更宽比特位算术逻辑。

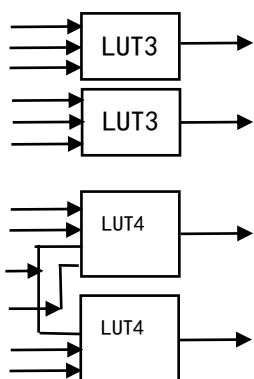
除普通逻辑模式外，PLB 还可以以 BANK 为单位，被配置为 Memory 模式，实现分布式 RAM 模式 (DRAM) 功能。如图二所示，当 BANK 被配置为 DRAM 时，同时占据 4 个 LUT 位置，进位链无法使用，但是 DFF 使用不受影响，且依然需要满足上述控制信号数目限制。

因为查找表为 2 输出设计，因此可以容纳多种输入总和不超过 6，输出数不超过 2 的复杂查找表排列组合。如图三所示，两个独立的 3 输入查找表，或有两个共享输入的 4 输入查找表，以此类推。

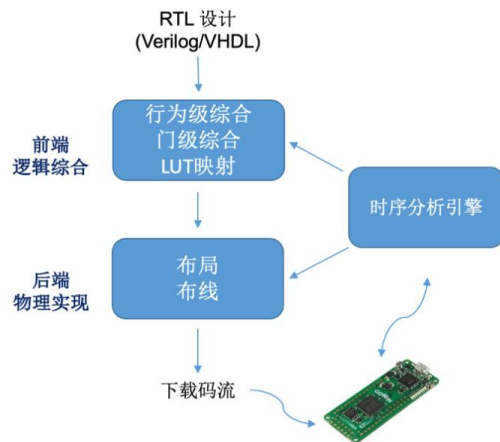
相同 PLB 内部的逻辑资源如 LUT 和 DFF，可以通过 PLB 内部的专用互连相互驱动；而跨 PLB 之间的连接，由固定长度为 x1,x2,x6 等全局互连相互驱动。



图二 PLB 详细结构; Logic 模式 (左), Memory 模式 (右)



图三 双输出 LUT6 配置示例



图四 FPGA EDA 流程

现代 FPGA 包含大量通讯协议 IP，处理器 IP 等预制单元。本赛题中，所有的 IP 接口，均被模拟为 IO，并在网表中采用统一模型 IPPIN 表示。

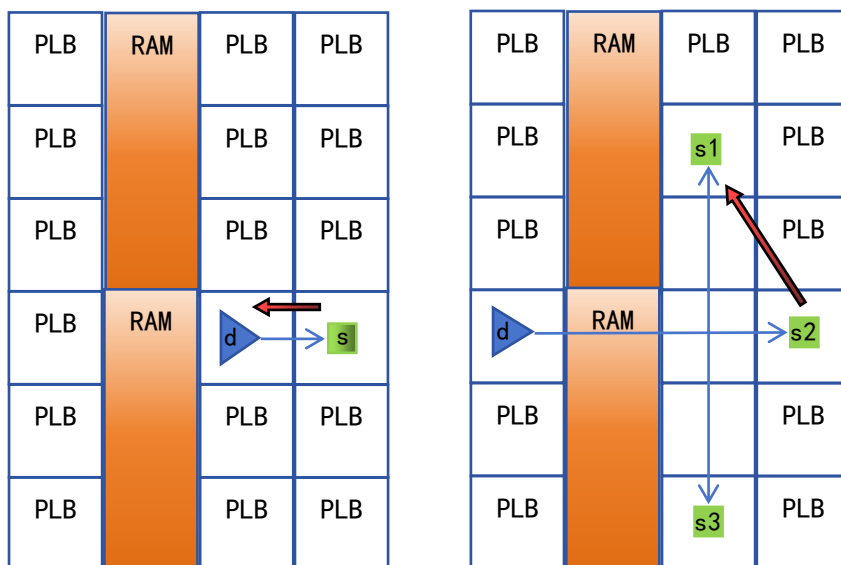
## 2、FPGA 布局与布线流程说明

与 ASIC 的 EDA 软件流程相似，FPGA 的 EDA 实现流程大体上包括了前端逻辑综合与后端物理实现两大步骤。前端逻辑综合步骤完成对用户输入 RTL 设计的行为级及与门级优化，再将优化后的门级电路网表映射在 FPGA 特有的基本逻辑单元 LUT 之上。物理实现步骤完成对综合后用户设计中各种类型逻辑单元与芯片物理资源之间的映射。物理实现步骤主要包括了布局，即根据资源类型与数量限制，为每个单元在芯片上找到合法的位置坐标。以及布线，即将布局好的单元通过合法的布线资源进行连接。

**问题来源：**随着器件整体规模的快速增长，布线问题已经逐渐成为 FPGA EDA 整体流程中最耗时的步骤。根据经验统计，布线在复杂的设计案例中 **runtime** 的占比会达到 **50%~70%**。因此优化布线 **runtime** 成为提高 EDA 收敛流程效率重要的环节。进一步分析布线耗时占比，算法处理跨 PLB 之间的网表复杂驱动关系造成的全局互连线冲突为主要耗时点。因此在布局阶段充分考虑未来布线需求已经成为整个 EDA 流程中必要的步骤。

解析式布局方法是目前工业界常用的集成电路布局方法之一。解析式布局可以从全局角度优化时序（线长）及控制单元密度。然而根据观察经验，在全局布局达到优化目标，完成单元摆放及合法化后，依然存在一些优化空间，可以在不扰动全局布局时序和单元密度的前提下，通过小范围的单元移动，进一步减少布线需求。

**问题示例：**如图五实例，驱动单元 **d** 及绿色扇出单元 **s**，如果通过短距离的移动，放置于和驱动 LUT 相同 PLB 单元中，则可以通过 PLB 内部互连相连接，节省全局布线资源。另一个实例为，当负载 **s1** 所在 PLB 仍有空余 LUT 位置的前提下，通过将多扇出线网另一个负载 **s2**，进行小范围移动，与负载 **s1** 布局在相同 PLB 单元内，增加了全局互连布线的共享，减少了整体布线需求。



图五 通过单元移动的详细布线优化

## 五、赛题描述：

本赛题总体优化目标为通过基本单元位置（LUT 及 DFF）移动或交换的手段减少整体布线线长。赛题输入为经过全局布局及合法化的电路网表。其中包含每个单元的初始位置，是否为可移动单元属性标记以及单元之间的连接关系。赛题输入还包含了一定比例（<10%）时序关键单元标记。详细输入文件格式描述请见附录及赛题网表 **Readme** 文件。

赛题输出为优化后网表单元位置。赛题不限制单元移动的整体距离及移动单元数目。但是优化后的网表需要遵守

1. 合法的资源布局。即与其位置相匹配的逻辑资源类型。不超过其位置所能容纳的资源上限。
2. 不得改变输入网表的连接关系。即不能增删改输入单元

及其之间 net 连接。不得通过合并复制等手段优化连接关系。

3. 输入标记为 **Fixed** 的单元须保持原始位置不得改变。

4. 满足上述的 **PLB** 内部 **LUT** 输入输出数目限制。

5. 满足上述的 **PLB** 内部控制信号上限限制。

6. 满足时钟区域内最大时钟信号数目限制。

7. 尽量不增加原始全局布局 **PLB** 单元 **pin** 密度分布。过高的 **pin** 密度会纳入质量评分考量范畴。

8. 时序关键路径线长尽量不恶化。

9. 适配赛会服务器的程序编译及运行脚本。

赛题评估环境：

1. 10 个练习网表及 10 个隐藏评分网表。

赛题将会提供 1~2 个小规模网表供调试练习使用。

其他赛题网表规模为：LUT 数目 50K~200K 区间

DFF 数目为 50K~300K 区间

2. 提供开放源码的检查及评分程序包含

(1) 合法化检查

(2) 基于曼哈顿距离及 FLUTE 算法[1]的线长评估

(3) PLB pin density 计算及报告

## 六、评分标准

1. 线长分数具体计算方式



线长的具体计算细分为 CriticalWireLength 及 Non-CriticalWireLength。其中 CriticalWireLength 计算方式为累计.timing 文件中标记的所有 timing critical pin 从 Driver 到 Sink 的曼哈顿距离之和。Non-CriticalWireLength 为 net 在除去 critical sink 之后，所计算出来的 FLUTE wirelength 总和。

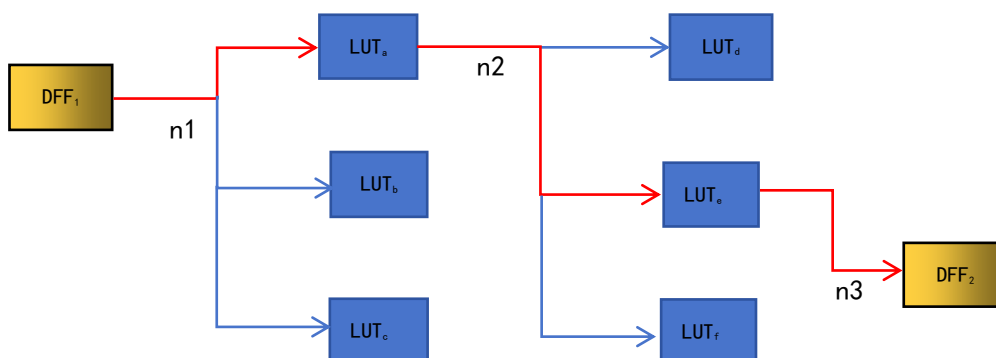
采用本计算方式的目的在于，希望通过单元移动和交换进行详细布局，对线长优化的过程中，对现有的小部分 critical path 线长尽量做到保护。即详细布局在不依赖时序分析的前提下，不破坏当前全局布局的时序关键路径。另外，时钟线网不参与整体线长计算，但是受上述芯片时钟数目限制。

图六为一条时序关键路径 DFF1→DFF2 及相关线网的线长计算实例。赛题提供的.timing 文件中，将会标记 DFF1 输出及 LUTa, LUTe, DFF2 输入 pin 为时序关键 pin。图中线网线长计算方式：

$wirelength_{n1} = DFF_1 \rightarrow LUT_a$  曼哈顿线长 + DFF<sub>1</sub> → 其他负载  
FLUTE 线长

$wirelength_{n2} = LUT_a \rightarrow LUT_e$  曼哈顿线长 + LUT<sub>a</sub> → 其他负载  
FLUTE 线长

$wirelength_{n3} = LUT_e \rightarrow DFF_2$  曼哈顿线长



图六 Critical Path 示意图

针对每一个测试案例，假设各个参赛组结果中最优线长为  $WL_{Best_{Crit}}$  及  $WL_{Best_{Non-Crit}}$

则线长得分  $L$  可以由以下公式计算得出

$$L = 100 - 100 * \left( \frac{WL_{crit}}{WL_{Best_{crit}}} - 1 \right) - 50 * \left( \frac{WL_{non-crit}}{WL_{Best_{non-crit}}} - 1 \right)$$

为防止优化线长造成的单元密集堆积。质量得分  $Q$  会综合考量密度指标并进行一定的惩罚。密度指标的量化考量方式为 PLB 连接的 pin 密度与 PLB pin 总数的比值。以本赛题 PLB 模型为例，8 个 6 输入 2 输出 LUT 及 16 个触发器共计 64 个输入 pin 及 32 个输出 pin。

$$P = \frac{\#connected\ input + \#connected\ output}{total\ input + total\ output}$$

假设原始布局结果 top 5% 可移动 PLB 平均 pin 密度为  $P$ ，优化后相对应的密度为  $P'$  且数值有所上升。则质量得分  $Q$  可以由以下公式计算得出

$$Q = \left(\frac{P}{P'}\right)^{2e} * L; \text{ if } P' > P$$

$$Q = L; \text{ if } P' \leq P$$

## 2. 程序质量评分

要求程序在合理运行时间和内存消耗下，正常运行结束，输出格式正确合法。

a) 赛题允许使用多线程实现，线程数上限为 8。

b) 运行时间上限：程序运行时间（wall time）超出 20 分钟，程序质量得分 Q 为 0 分。程序运行时间统计，届时会根据服务器环境，运行多次取平均值。

c) 运行结果稳定性：要求多次运行，在输入条件保持不变的前提下，输出结果相同。对多次运行结果不稳定的网表结果，质量得分 Q 为 0 分

d) 运行内存上限：运行内存超过 15G，程序质量得分 Q 为 0 分。

e) 合法 case 结果通过计算公式获得  $Q_{case}$

为合并 10 个考核案例质量得分 Q，并取平均值，

$$Q_{final} = \frac{\sum Q_{case}}{10}$$

## 七、附录

赛题输入文件包括：

1. **arch** 文件，FPGA 芯片资源描述。具体包含 **lib**, **scl** 及 **clk** 三个描述文件。包含 **M** 行 **N** 列的阵列。阵列包含 **PLB**, **RAM**, **DSP**, **IO** 及 **IP** 等各种资源。给定每个 **x**, **y** 坐标系下，阵列资源类型。给定若干时钟区域范围，及每个时钟区域内最大时钟信号数目上限。如第二节描述，每个 **PLB** 资源数量限制为 2 个 **bank**，每个 **bank** 中包含 4 个 6 输入 2 输出查找表及 8 个 **DFF** 单元。每个 **bank** 中的 **DFF** 限制必须连接相同的 **clock** 及 **reset** 信号，每个 **bank** 最多能容纳两个不同的 **ce** 信号。详细信息请参考 Arch 格式说明文件。

2. **\*.node** 文件，设计网表单元描述。网表单元类型包含查找表 **LUT**，触发器 **DFF**，进位链，**RAM**，**DSP**，**IO** 等多种逻辑资源类型。**.node** 文件提供了每一个单元的资源类型及当前位置。以及是否可以移动的属性标记。

3. **\*.net** 文件，设计网表连接关系描述。提供线网属性标记，包含 **clock**，**control** 等。电路网表已完成了合法化步骤，因此所有输入网表，满足上述时钟资源及控制资源限制。

4. **\*.timing** 文件，时序关键 **pin** 标记。选取网表中~10%的 **pin** 标记为时序关键 **pin**，希望在优化过程中得到一定程度的保护。即保持 **driver** 到时序关键的 **pin** 曼哈顿距离不变或有所缩短。

## 八、参考文献

[1] FLUTE: Fast Lookup Table Based Technique for RSMT Construction and Wirelength Estimation

<https://www.engineering.iastate.edu/~cnchu/flute.html>

\*本赛题指南未尽问题，见赛题 Q&A 文件