# 超图分割算法设计

## 一. 作业背景

随着集成电路设计的不断发展，硬件仿真逐渐成为了现代电路设计中不可或缺的重要环节。相比软件仿真，硬件仿真能够显著提升仿真速度，尤其在处理大规模电路设计时更具优势。然而，随着电路规模的快速增长，单个FPGA（现场可编程门阵列）的容量变得有限，无法满足高性能设计的需求。因此，为了应对这种挑战，需要使用多FPGA系统（MFS）来处理这些大规模的工程。在多FPGA系统中，系统分割算法的质量和速度成为了关键的挑战。一个优秀的分割算法能够在可控的时间内，生成对时序优化有利且易于FPGA内部布局和布线的分割结果。

## 二. 作业目标

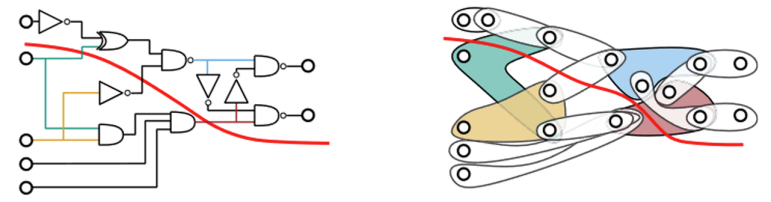
本次作业的目标是设计和实现一个高效的系统分割算法，在满足**FPGA hop**，**资源**以及**对外互联数**三个约束条件下，优化指定的目标函数，最终输出合理的划分结果。

## 三. 作业流程

### 步骤一：输入

输入总共四个文件：design.info，design.topo，design.net和design.are。其中design.net和design.are是对于电路网表的描述，design.info和design.topo是对于fpga的描述。

电路网表的描述



对于一个电路设计，我们首先需要将电路网表映射为**超图**。以上图为例，具体做法是：

* 将门单元表示为节点
* 将线网表示为超边（连接多个顶点的集和）

电路网表即可以表示为超图，方便后续利用超图划分算法进行划分。

本次作业的电路网表有如下**特性**：

* 电路网表中的每个门单元，会使用**FPGA资源**（共考虑FF、LUT、BUFG、TBUF、DCM、 BRAM、DSP、PP八种资源）。
* 线网表示成的超边是 **加权有向超边**（超边里的节点分为**驱动节点**和**被驱动节点**两类）。

在design.are和design.net中分别描述了这三个特性：

* design.are：描述每个节点使用的资源量。每行一个节点，后面8个数字分别表示8种资源使用量。

**g1 9 0 2 0 0 0 0 0  
 g2 5 0 1 0 0 0 0 0  
 g3 4 0 0 0 0 0 0 0  
 g4 1 0 0 0 0 0 0 0**

* design.net：描述每个超边的权重以及它包含的节点。每行一个超边，第一个节点为驱动节点， 后面的数字表示超边权重，后续若干节点表示被驱动节点。

**g4 9 g1  
 g4 10 g3  
 g2 3 g1 g4**

FPGA的描述

本次作业的FPGA有如下**特性**：

* FPGA的**互联**：FPGA之间并不是两两都互联的，信号只能在直接互联的FPGA之间传递。
* FPGA的**资源**：共考虑八种资源：FF、LUT、BUFG、TBUF、DCM、 BRAM、DSP、PP。
* FPGA的**对外互联数**：包含位于这个FPGA上的节点且被割超边的权重总和。

在design.info和design.topo中分别描述了这三个特性：

* design.info：描述每个FPGA的最大对外互联数与可用资源量。每行一个FPGA，第一个数字为它的最大对外互联数，后面8个数字分别表示8种资源的最大可用量。

**FPGA1 100 10 0 2 0 2 0 2 4**

**FPGA2 100 10 2 2 0 2 0 2 4**

* design.topo：描述FPGA之间的连线以及最大hop距离。第一行一个数字表示允许的最大hop距离。第二行表示给出的两个FPGA之间存在连线。

**2**

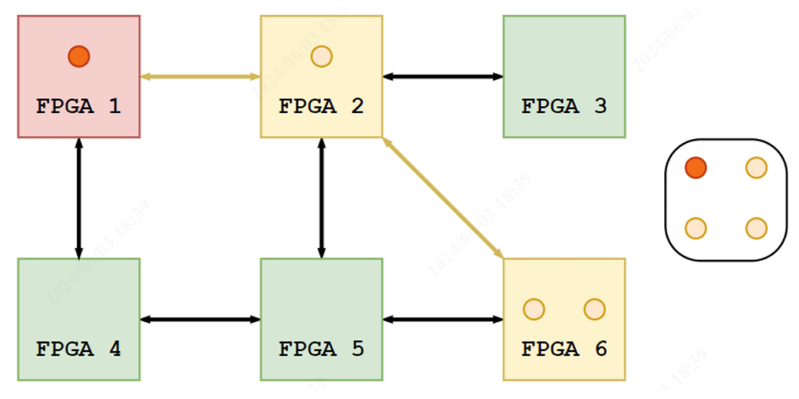
**FPGA1 FPGA2**

### 步骤二：超图划分

超图划分的工作就是将电路网表的节点和超边合理地映射到FPGA资源上。接下来讲解本次作业的**划分要求**。

优化目标

本次作业的优化目标是**最小化总的hop路径长度**。Hop路径的长度定义为信号从源FPGA到目标FPGA所需经过的最少中转FPGA数量加1。优化的意义在于，信号经过FPGA间的外部连线传输时，延时会显著增加，因此减少hop路径能够有效降低系统的延时。以下图为例：



* 超边的权重是3，其中有4个节点，红色的为驱动节点，其余为被驱动节点。 驱动节点放在FPGA1上，一个被驱动节点放在FPGA2上，剩下两个节点放在FPGA6上。
* 此时产生了两条路径，FPGA1 -> FPGA2与FPGA1 -> FPGA6，他们的hop路径长度分别为 1 与 2。
* 这条超边带来的hop路径长度总共为3∗(1+2)=9。
* **总hop路径长度**即所有超边带来的hop路径长度总和。

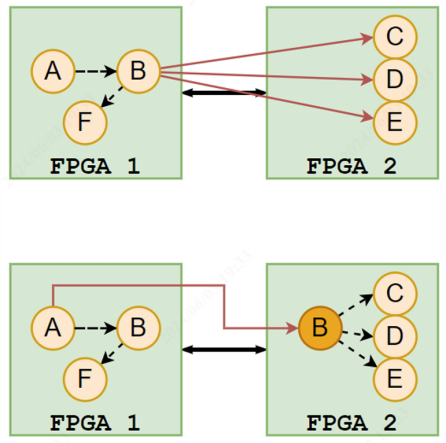
约束限制

本次作业共如下三个约束限制：

* **资源限制约束**：分割方案需要保证每一个FPGA上的每一种资源的使用量不超过上限。
* **hop约束**：整个多FPGA系统的延时取决于最大延时，因此要求分割方案的最大hop距离不能超过给定上限。
* **对外互联约束**：当一个FPGA 的对外信号数过多时，会导致FPGA 内部布线拥塞，进而导致编译失败。因此要求每个FPGA的对外互联数要小于给定数值。

拓展项

通过将电路中的逻辑单元复制到多个需要的FPGA上产生信号会带来更好的系统性能，其代价是资源使用量的增加。例如下图所示，此时存在三条边被切割。如果将节点B移动到FPGA 2中，又会带来两条新的边被切割。但如果将节点B复制一份到FPGA2中，则可以只有一条边被切割，从而以资源使用量增加为代价优化了系统性能。



逻辑复制功能当做拓展功能，能够进一步优化**总hop距离**。在约束满足情况下，本次作业不限制复制的节点数。

### 步骤三：输出

输出文件是design.fpga.out，其中每一行表示一个FPGA，后续跟若干节点，表示被划分到这个FPGA的所有节点，被复制出的新节点用“\*” 标识。

**FPGA1: g2 g3 g4**

**FPGA2: g1 g4\***

## 四. 评测标准

* 每组需要提交可执行文件（命名为partitioner），如果使用了动态库，需要包含相关运行所需文件。
* 可执行文件需要满足如下输入输出要求：

使用-t参数指定测试数据路径，测试路径下包含四个输入文件。

使用-s参数指定输出文件design.fpga.out路径。

**./build/bin/partitioner \**

**-t ./data/case01 \**

**-s ./design.fpga.out**

测试数据在项目文件夹/data下，命名为case加数字，共四个测试case。

* 评估器使用-t参数指定测试数据路径，使用-s参数指定输出的design.fpga.out文件路径，其功能是评估design.fpga.out结果是否合法，并给出总hop数来评断分割器的效果。

**./etc/evaluator \**

**-t ./data/case01 \**

**-s ./design.fpga.out**

* 单个case运行时间限定在一小时以内。

详细情况参考项目的scripts脚本和README.md说明。

## 五. 样例解读

输入输出

输入总共有四个文件，分别是：

* design.info：

**FPGA1 100 10 0 2 0 2 0 2 4**

**FPGA2 100 10 2 2 0 2 0 2 4**

* design.are：

**g1 9 0 2 0 0 0 0 0**

**g2 5 0 1 0 0 0 0 0**

**g3 4 0 0 0 0 0 0 0**

**g4 1 0 0 0 0 0 0 0**

* design.net：

**g4 9 g1**

**g4 10 g3**

**g2 3 g1 g4**

* design.topo：

**2**

**FPGA1 FPGA2**

输出文件是：

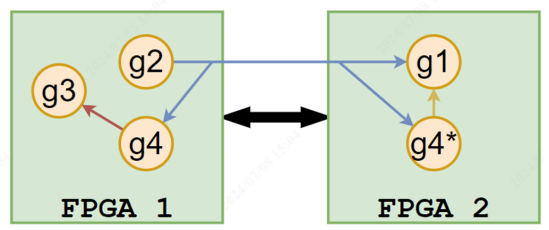
* design.fpga.out：

**FPGA1: g2 g3 g4**

**FPGA2: g1 g4\***

输出方案解读

design.fpga.out的分割方案可视化如下图，三条超边用三种颜色。



* 约束条件

1. FPGA1 上资源总使用：10 0 1 0 0 0 0 0 FPGA2 上资源总使用：10 0 2 0 0 0 0 0 均资源使用量均不超过可用资源。
2. 每一个节点都出现在输出文件中。
3. FPGA1对外互联3 ；FPGA2对外互联3，均小于100。
4. 只有一条hop路径，它的长度为1，小于2。

综上，满足了所有约束目标，该方案是合理的方案。

* 优化目标计算

被切割的超边权重为3，hop只有一条hop距离为1的路径，因此目标函数值为3\*1=3。