2025 中国研究生创“芯”大赛·EDA 精英挑战赛

一、赛题名称

支持重新组网的多 FPGA 系统布线算法设计

二、命题单位

上海思尔芯技术股份有限公司

三、赛题背景

随着 IC 设计规模越来越大，硬件仿真平台难以用单个或极少量的 FPGA 仿真高达数十亿门规模的设计。因此，我们需要应用大规模的多 FPGA 系统 (Multi-FPGA System, MFS)处理此类工程。在 MFS 中，大量的信号需要在 FPGA 之间传递。由于硬件结构的限制，并不是所有的 FPGA 之间都能直接进行互联通信，因此需要根据实际的硬件互联拓扑规划信号的通信路径，此过程被称为系统级布线。

FPGA 的系统级布线是一个具有挑战性的问题，因为布线资源相对稀缺。这可能会导致由于绕开拥塞而造成布线路径过长，从而使性能变差，或者导致某些信号无法完成布线。

FPGA 的布线问题可以简单表述为：将信号分配到布线资源上，在实现所有信号成功连接的同时优化整体性能。第一个目标，即完成所有信号的布线。第二个目标，为信号选择延迟最小的路径。在布线过程可以对 FPGA 组网进行一定程度的调整来优化路径延时。

四、赛题描述

本赛题属于数字电路领域。处在整个 EDA 流程的验证与仿真阶段。具体描述为给定一个分割好的逻辑网表（Netlist）和一个多 FPGA 平台拓扑结构，如何将所有跨

FPGA 的信号连接映射到物理互连资源上，使得满足所有连接需求，并在资源和延迟约束下最小化最大路径延迟。

本赛题会用到的符号说明见表 1。

表 1：符号说明

符号	解释说明
$G=(V,E)$	无向加权图，V是点集，E是边集
$w(F_i, F_j)$	F_i与F_j之间的连接通道数量
n_i	网表中的一个网络，描述了一组逻辑单元之间的连接关系
s_i	net n_i的源端
t_{ij}	net n_i的一个接收端，其源端对应的是s_i
P_{ij}	从源端s_i到接收端t_{ij}的路由路径
$D(P_{ij})$	路径P_{ij}的总延迟
N_{pq}	布线过程中跨越F_p与F_q之间的通道的 net 集合
r_{pq}	有连接通道的一对 FPGA F_p和F_q之间的 TDM 比率
$D(r_{pq})$	TDM 比率r_{pq}对应的 TDM 延迟
$D(n_i)$	net n_i的延迟，为其所有源到接收端路径延迟的最大值
$\delta_{(F_p, F_q)}$	F_p与F_q之间连接通道变动量

由于 FPGA 之间物理互连通道数量有限，为实现跨 FPGA 的高带宽信号传输，通常在 FPGA 边界引入 TDM（时分复用，Time-Division Multiplexing）IP 模块，以对信号进行时分复用处理，从而在逻辑上动态扩展物理通道的传输能力。

对于任意一条物理连接通道，若通过该通道传输的逻辑信号数为r，则称该通道的 TDM 比例（TDM Ratio） 为r。换言之，TDM Ratio 表示该通道在一个时分复用周期内可承载的独立信号数量。我们也可等效地理解为：通过该通道传输的每个信号，其对应的 TDM Ratio 等于该通道的 TDM Ratio。

在本赛题中，每一对 FPGA 之间的所有物理连接通道均采用统一的 TDM Ratio。

输入：

i.一个多 FPGA 平台拓扑结构

表示为一个无向加权图$G = (V, E)$。如图 1 所示，其中顶点集合V表示系统中所有的 FPGA 的集合，边集合E表示 FPGA 之间的物理连接通道，这些通道支持时分复用(TDM，Time-Division Multiplexing)I/O。

每条边$e \in E$连接两个 FPGA，表示它们之间存在一组可用的物理通信链路。我们将边的权重$w(e)$ 定义为该对 FPGA 之间物理通道的数量。换言之，边的权重反映了两个 FPGA 间的通信资源容量。例如，如图 1 所示，FPGA F_1与 FPGA F_2之间存在三条独立的物理通道，则其对应边的权重为$w(F_1, F_2) = 3$。

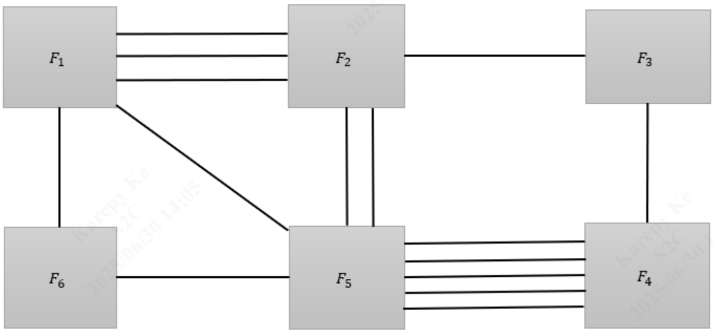

图 1 多 FPGA 拓扑结构示意图

ii.一个分割好的超图

表示为一个有向超图$H = (V, N)$。其中顶点集合V与上文中表示 FPGA 集合的顶点一致，超边集合$N = \{n_1, n_2, \ldots, n_k\}$为 net 的集合，每个 net n_i是一个两端或多端的连接，表示为包括源端s_i和若干个接收端t_{ij}在内的终端集合，每个 net 中的端点被映射到了不同 FPGA 上。n_i表示为V的一个子集。如图 2 所示，包含三个不同的 net：

- $n_1 = \{s_1, t_{11}, t_{12}\}$
- $n_2 = \{s_2, t_{21}\}$

- $n_3 = \{s_3, t_{31}, t_{32}\}$

其中，s_i表示通信的源端，t_{ij}为对应的接收端，所有端点均映射至不同的 FPGA 上。

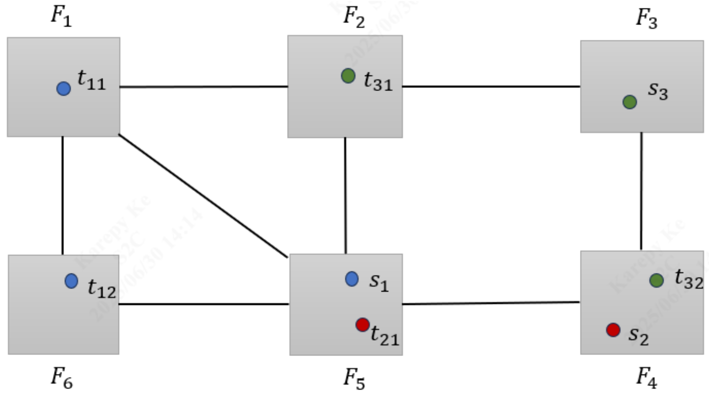

图 2 nets 端点映射示意图

输出：

对于每个 net n_i ，目标是在多 FPGA 平台的互联图中分配一条路径或多条路径，使得其源端s_i与所有接收端t_{ij}之间通过跨越一个或多个 FPGA 的路径实现有效连接。

延迟模型：

设从源端s_i到接收端t_{ij}的路由路径为P_{ij}，该路径的跨 FPGA 总延迟记为$D(P_{ij})$。路径延迟由以下两个部分构成：

1.TDM 传输延迟：路径上每对相邻 FPGA 之间的时分复用带来的延迟。对于 FPGA F_p和 F_q之间的通信，其 TDM 延迟为

$$D(r_{pq}) = \alpha * r_{pq} \qquad (1)$$

其中r_{pq}表示该连接的 FPGA 之间 TDM 比率，定义为$r_{pq} = \frac{|N_{pq}|}{w(F_p,F_q)}$，其中$N_{pq}$为布线过程中跨越该通道的 net 集合；α 为给定系数（本赛题中设定为 α ＝0.7）。

2.布线延迟：每跳跨 FPGA 的物理互连带来的固定延迟，记为常数 β（本赛题中设定为 β ＝30）。

因此，总路径延迟 $D(P_{ij})$ 是路径上各跳延迟的累加，包含上述两部分。

Net 延迟定义：

每个 net n_i的延迟定义为其所有源到接收端路径延迟的最大值，即：

$$D(n_i) = \max_{1\leq j<|n_i|} D\left(P_{ij}\right) \qquad (2)$$

其中$|n_i|$表示 net n_i中终端的总数。

优化目标：

本赛题的优化目标是最小化所有 net 中最大路径延迟，即：

$$\text{minimize}\left(\max_{1\leq i\leq k} D(n_i)\right) \qquad (3)$$

如图 3 所示，图中的蓝色、红色与绿色路径分别表示为 nets n_1, n_2, n_3所分配的跨 FPGA 路由路径。

以n_1为例，其包含两个从源端到接收端的路径：

- 路径 $P_{11} = F_5 \rightarrow F_2 \rightarrow F_1$
- 路径 $P_{12} = F_5 \rightarrow F_6$

路径 P_{11} 跨越两个 FPGA 边界，因此其总延迟为：

$$D(P_{11}) = D(r_{52}) + D(r_{21}) + 2 * \beta \qquad (4)$$

其中：

- $D\left(r_{pq}\right) = \alpha * r_{pq}$为 FPGA F_p与F_q之间的 TDM 延迟；
- β 为每次跨 FPGA 跳转所引入的固定布线延迟。

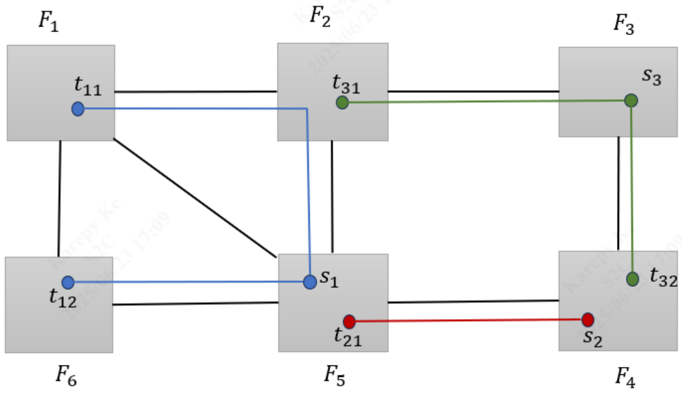

图 3 布线结果示意图

以下给出**示例说明 TDM ratio** 的计算以及 **net 跨通道统计规则**：

如图 4 所示，考虑$\mathrm{n}_1, \mathrm{n}_2, \mathrm{n}_3, \mathrm{n}_4$的布线情况：

- FPGA 对之间的物理通道数量为：

$$w(F_3, F_4) = 2, w(F_4, F_5) = 2, w(F_5, F_6) = 1 \qquad (5)$$

- 实际布线穿越这些通道的 net 数量为：

$$|N_{34}| = 2, |N_{45}| = 4, |N_{56}| = 2 \qquad (6)$$

因此，三对 FPGA 之间的 TDM 比率分别为：

$$r_{34} = \frac{2}{2} = 1, r_{45} = \frac{4}{2} = 2, r_{56} = \frac{2}{1} = 2 \qquad (7)$$

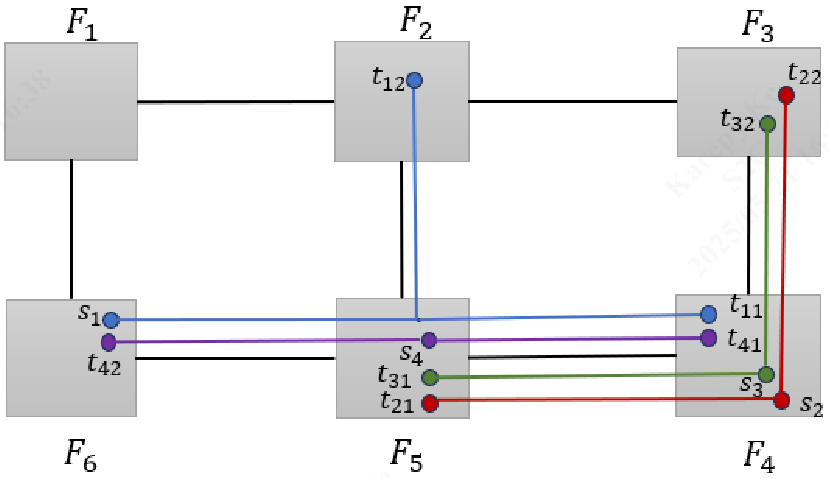

图 4 布线结果示意图

Net 跨通道统计规则说明：

特别需要指出的是，在统计跨通道 net 数量时，同一个 net 多次经过某一对 FPGA 连接通道，仅计为一次。例如，如图 4 所示，蓝色 net n_1从源端到两个接收端分别对应如下两条路径：

- $P_{11} = F_6 \rightarrow F_5 \rightarrow F_4$
- $P_{12} = F_6 \rightarrow F_5 \rightarrow F_2$

虽然这两条路径均包含通道$F_6 \rightarrow F_5$，但在计算布线穿越该通道的 net 数量$|\mathrm{N}_{65}|$时，net n_1仅计入一次，以避免重复统计造成的资源高估。

在布线过程中，为进一步优化路径延迟，可以考虑**动态调整 FPGA 之间的物理互联拓扑结构**，即在现有任意 FPGA 对之间**增加、删除或交换物理连接通道**，以达到改善关键路径延迟的目的。

组网调整示例与路径优化：

如图 5 所示，设当前所有相邻的 FPGA 之间的物理连接通道数量均为 1。在该组网下，最大路径延迟来自 net n_1的路径$P_{11} = F_5 \rightarrow F_2 \rightarrow F_1$，其延迟为：

$$D(P_{11}) = D(r_{52}) + D(r_{21}) + 2 * \beta \qquad (8)$$

观察可知，FPGA F_5与 F_2之间的通道仅被 net n_1使用，若将该物理通道重新分配至 FPGA F_5与F_1之间，不会影响其他 nets（如$\mathrm{n}_2, \mathrm{n}_3$）的连接。

此时，路径P_{11}可更新为更短的路径：

$$P_{11} = F_5 \rightarrow F_1, D(P_{11}) = D(r_{51}) + \beta \qquad (9)$$

该调整显著减少了最大路径延迟，从而优化了系统整体性能。

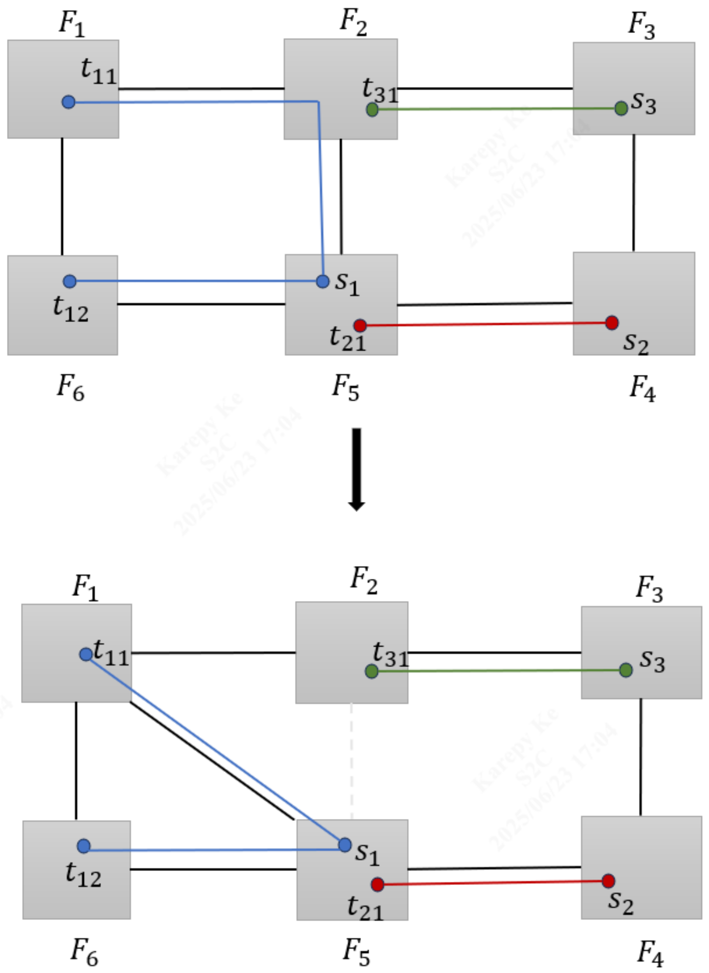

图 5 调整组网示意图

通道变动量说明：

为刻画组网调整对物理资源的影响，引入**连接通道变动量**的定义。设 $\delta_{(\mathrm{F_p,F_q})}$ 表示 FPGA $\mathrm{F_p}$与$\mathrm{F_q}$之间的物理连接通道的改变量：

- $\delta_{(F_p,F_q)} > 0$：表示新增 FPGA $\mathrm{F_p}$与$\mathrm{F_q}$之间的物理通道
- $\delta_{(F_p,F_q)} < 0$：表示删减已有的通道
- $\delta_{(F_p,F_q)} = 0$：表示该通道保持不变

在上述示例中，组网调整如下：

$$\delta_{(F_5,F_1)} = +1, \delta_{(F_5,F_2)} = -1 \qquad (10)$$

其余 FPGA 对之间的改变量均为 0。因此，此次拓扑重构的总通道变动量为：

$$\sum_{(F_p,F_q)} \left|\delta_{(F_p,F_q)}\right| = 2 \qquad (11)$$

五、评分标准

在物理组网调整与布线优化过程中，需满足以下约束条件：

1.TDM 比率约束

对于任意一对相连的 FPGA $\mathrm{F_p}$与$\mathrm{F_q}$，其时分复用比率$\mathrm{r_{pq}}$不得超过系统设定的最大允许值$\mathrm{R_{max}}$，即：

$$r_{pq} \le R_{max}, \forall\left(F_p, F_q\right) \in E \qquad (12)$$

2.TDM 比率对齐约束

所有 TDM 比率必须为 8 的整数倍，以满足 IP 模块对时分复用配置的硬件要求，即：

$$r_{pq} \in \{8, 16, 24, \dots\}, \forall\left(F_p, F_q\right) \in E \qquad (13)$$

3. FPGA 连接通道容量约束

任意 FPGA $\mathrm{F_p}$ 的对外连接通道总数不得超过设定上限$\mathrm{W_{max}(F_p)}$。设 FPGA $\mathrm{F_p}$ 的对外连接通道总数为：

$$\sum_{F_q \in V, p \ne q} w\left(F_p, F_q\right) \le W_{max}\left(F_p\right), \forall F_p \in V \qquad (14)$$

4.通道调整范围约束

所有连接通道变动量的绝对值总和，即：

$$\sum_{(F_p,F_q)\in(V\times V)} \left|\delta_{(F_p,F_q)}\right| \qquad (15)$$

不得超过当前组网中物理通道总数的 30%。设通道总数为 $\mathrm{W_{total}}$，则有：

$$\sum_{(F_p,F_q)\in(V\times V)} |\delta_{(F_p,F_q)}| \le 0.3\cdot W_{total} \qquad (16)$$

例如，在图 5 中，物理通道总量为 7，调整后的通道改变量总和为 2，满足该约束。

运行时间限制：为了不卡评测服务器，本赛题会给出运行时间限制。需要保证在所有的测例中，运行时间不能超过 **1 小时**（该运行时间包括数据读入及结果输出耗时）。在结果合法的前提下，算法的运行时间也是评判的标准之一。**本赛题不支持多线程。**

运行内存限制：需要保证在所有的测例中，峰值内存消耗小于 **32GB**。

计分细则：对于每个 case，本赛题会使用评估工具运行选手提供的可执行文件，并对结果进行打分。我们将对所有分数进行排序并赋分。假设共有 20 支队伍参赛，第一名将获得 20 分，第二名 19 分，以此类推。最终将所有 case 的得分加权相加，作为最终得分。对于每个 case，它的分数越小则排名越靠前。此分数包含两部分，假设最大延迟为x，程序运行时间为y秒，则得分为

$$score = x * \left(1 + 0.2 * \frac{y}{3600}\right) \qquad (17)$$

若结果不合法或运行时间超过 1 小时，则在此 case 中排名为最后一名。

参赛选手需要提交满足上述要求的可执行文件（如果使用了动态库，需要包含相关运行所需文件），并确保其可以在竞赛服务器上正确运行。

六、测例情况

赛题定义节点数目为所有超边上节点数目的总和，即：

$$节点数目 = \sum_{e\in E} |e| \qquad (18)$$

所有测例的规模情况如下表所示，其中公开测例的打分权重低于隐藏测例。

表 2 测例情况表

编号	状态	权重	数据规模
1	公开	0.6	节点数目<=100，FPGA 数目<=4
2			节点数目<=1W，FPGA 数目<=8
3			节点数目<=20W，FPGA 数目<=32
4			节点数目<=500W，FPGA 数目<=64
5-6	隐藏	1.0	节点数目<=1W，FPGA 数目<=8
7-8			节点数目<=20W，FPGA 数目<=32
9			节点数目<=100W，FPGA 数目<=64
10			节点数目<=500W，FPGA 数目<=64

七、输入输出文件说明

输入文件：

- **design.info：**

描述系统中每个 FPGA 节点的资源约束，具体为其允许的最大对外连接通道数量。

- **design.net：**

描述系统的通信网络。每一行为一个超边（net），包含该 net 的权重（如通信强度或带宽需求）及其关联的多个逻辑节点（源端和接收端），构成一个有向超图。

- **design.topo：**

描述当前物理拓扑结构中 FPGA 之间的物理连接关系。每一对 FPGA 的连接通道信息由该文件给出，定义了初始拓扑图$G = (V，E)$。

- **design.fpga.out：**

描述逻辑节点到物理 FPGA 的映射结果，即超图分割结果。每个逻辑节点所属的 FPGA 编号在该文件中给出，为后续布线阶段提供位置基础。

输出文件：

- **design.route.out：**

表示布线阶段的输出结果，给出每个 net 中源端与所有接收端之间的具体路径。路径以跨越 FPGA 节点序列的方式表示，反映了信号实际传输路径。

- **design.newtopo：**

表示组网调整后的物理互连结构。该文件记录了在优化过程中对初始拓扑进行的增删改操作后形成的新拓扑图，包括各对 FPGA 之间更新后的连接通道信息。

八、样例

输入样例（图 6）

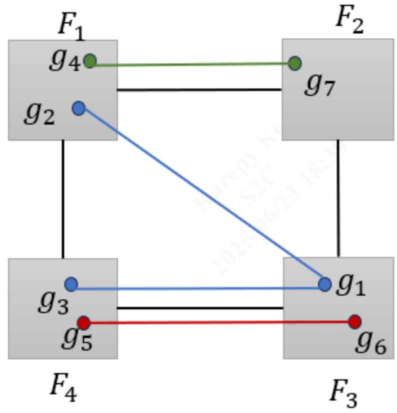

图 6 输入样例示意图

design.info 文件格式说明（资源约束）：

该文件描述系统中各个 FPGA 的物理连接资源约束。文件按行组织，每行对应一个 FPGA 节点，其格式如下：

<FPGA_ID> <Max_IO>

- <FPGA_ID>：FPGA 的唯一标识符（例如 F1、F2 等）。
- <Max_IO>：该 FPGA 节点允许的最大对外连接通道数量，表示其 I/O 接口的资源上限。

例如，以下示例中表示系统中共有 4 个 FPGA，每个 FPGA 的最大对外连线数为 3：

```
F1 3
F2 3
F3 3
F4 3
```

design.net 文件格式说明：

该文件描述系统的通信需求网络，采用超图形式表示逻辑信号连接关系。文件按行组织，每行对应一个 net（超边），格式如下：

<Source_Node> <Weight> <Sink_Node_1> <Sink_Node_2> ...

- <Source_Node>：该 net 的源端节点（source terminal）。
- <Weight>：该 net 的权重，表示通信需求的强度或优先级。本赛题中所有 net 的权重统一设置为 1。
- <Sink_Node_i>：该 net 的一个接收端节点（sink terminal）。每行可包含多个接收端。

例如，以下示例表示三个 net：

```
g1 1 g2 g3
g4 1 g7
g5 1 g6
```

design.topo 文件格式说明（初始物理拓扑）：

该文件用于描述系统中各 FPGA 节点之间的初始物理连接拓扑，采用邻接矩阵形式编码连接通道数量。

文件按行组织，每行对应一个 FPGA 节点，其格式如下：

<FPGA_ID>: <C_1>, <C_2>, ..., <C_n>

- <FPGA_ID>：当前行所描述的 FPGA 节点编号。
- <C_i>:表示当前 FPGA 与编号为第 *i* 个的 FPGA 节点之间的物理连接通道数量。
- 数值顺序按照所有 FPGA 的索引排序,包括自身(即对角线元素,通常为 0)。

本格式等价于一个n×n的对称邻接矩阵（如连接为双向），其中第 i 行表示 FPGA F_i 与所有 FPGA（包括自己）之间的连接通道数量。

示例如下（图 6）：

F1: 0,1,0,1

F2: 1,0,1,0

F3: 0,1,0,1

F4: 1,0,1,0

design.fpga.out 文件格式说明（逻辑分割结果）：

该文件记录逻辑节点与物理 FPGA 之间的映射关系，即超图分割后的节点分配结果。该文件每行对应一个 FPGA，列出所有被分配到该 FPGA 的逻辑节点。

其格式如下：

<FPGA_ID>: <node_1> <node_2> ... <node_3>

示例如下（图 6）：

F1: g2 g4

F2: g7

F3: g1 g6

F4: g3 g5

上述示例中，逻辑节点 g2 与 g4 被分配到 FPGA F1，g7 被分配到 F2，以此类推

输出样例（图 7）

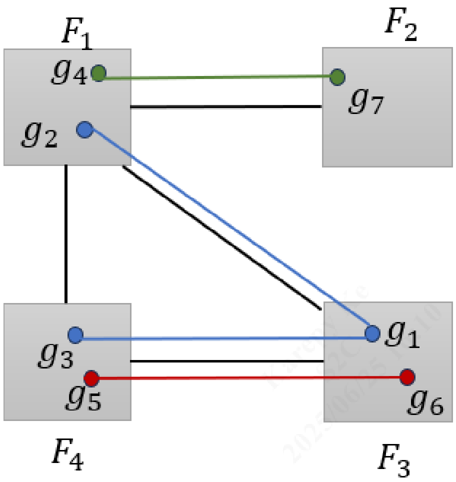

图 7 输出样例示意图

design.route.out（布线结果文件）：

该文件记录每条逻辑 net 的具体布线路径信息。文件按 net 的路径延迟从高到低排序，每个 net 占据若干行，格式如下：

[Net ID]

[FPGA sequence to Sink 1] [path delay]

[FPGA sequence to Sink 2] [path delay]

…

- [Net ID]：逻辑 net 的编号（从 1 开始），对应于 design.net 中的行序号。
- [FPGA sequence to Sink]：从该 net 的源端所在 FPGA 到某个接收端所在 FPGA 的布线路径，表示为一串按顺序排列的 FPGA 编号索引。

- [path delay]：该路径的累计延迟，计算方式为跨 FPGA 跳数乘以布线延迟和 TDM 延迟之和。

示例说明（图 7）：

[net 1]

[3,4] [35.6]

[3,1] [35.6]

该示例表示 Net 1（对应图 7 中的蓝色 net）具有两个接收端：

- 从源端 FPGA F3（索引 3）到第一个接收端 FPGA F4（索引 4）的路径为 F3 → F4，路径延迟为 30+0.7*8=35.6；
- 到第二个接收端 FPGA F1（索引 1）的路径为 F3 → F1，路径延迟为 30+0.7*8=35.6。

所有 nets 将按其最大路径延迟$D(n_i)=\max_j D(P_{ij})$ 进行降序排列，以突出关键路径。

design.newtopo（拓扑调整后物理互联结构）：

该文件表示在布线优化过程中对 FPGA 间连接关系进行调整后的结果。其格式与输入文件 design.topo 一致，采用邻接矩阵的方式描述更新后的连接通道数量。

每行对应一个 FPGA 节点，格式为：

<FPGA_ID>: <C_1>, <C_2>, ..., <C_n>

示例说明（图 7）：

F1: 0,1,1,1

F2: 1,0,0,0

F3: 1,0,0,1

F4: 1,0,1,0

九、参考资料

[1] Lin T W, Tai W C, Lin Y C, et al. Routing topology and time-division multiplexing co-optimization for multi-FPGA systems[C]//2020 57th ACM/IEEE Design Automation Conference (DAC). IEEE, 2020: 1-6.

[2] Zou P, Lin Z, Shi X, et al. Time-division multiplexing based system-level FPGA routing for logic verification[C]//2020 57th ACM/IEEE Design Automation Conference (DAC). IEEE, 2020: 1-6.

[3] Turki M, Marrakchi Z, Mehrez H, et al. Iterative routing algorithm of inter-FPGA signals for multi-FPGA prototyping platform[C]//International Symposium on Applied Reconfigurable Computing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013: 210-217.

[4] Khalid M A S, Rose J. A novel and efficient routing architecture for multi-FPGA systems[J]. IEEE transactions on very large scale integration (VLSI) systems, 2000, 8(1): 30-39.

[5] McMurchie L, Ebeling C. PathFinder: A negotiation-based performance-driven router for FPGAs[C]//Proceedings of the 1995 ACM third international symposium on Field-programmable gate arrays. 1995: 111-117.

[6] (书) 超大规模集成电路物理设计 从图分割到时序收敛

*本赛题指南未尽问题，见赛题 Q&A 文件

2025 China Postgraduate IC Innovation Competition・EDA Elite Challenge Contest

1. Problem

Routing Algorithm Design for Multi-FPGA Systems Supporting Topology Reconfiguration

2. Company

Shanghai S2C Technology Co., Ltd.

3. Problem Background

As the scale of IC designs continues to grow, modern hardware emulation platforms face increasing challenges in simulating designs with gate counts in the billions using only a single or a small number of FPGAs. To address this, Multi-FPGA Systems (MFSs) have become essential for handling such large-scale verification tasks.

In an MFS, a massive number of signals must be transmitted across different FPGAs. Due to architectural constraints, not all FPGAs are directly interconnected, which necessitates careful planning of signal routing paths based on the actual hardware interconnect topology — a process known as system-level routing.

System-level FPGA routing is a challenging problem due to the limited availability of interconnect resources. Routing congestion may force signals to detour along longer paths, resulting in degraded performance. In some cases, certain connections may become unroutable due to insufficient resources.

The core problem can be abstracted as follows: assign each signal to available interconnect resources, such that all connectivity requirements are satisfied while optimizing overall system performance. The primary objective is to ensure routability for all signals. The secondary objective is to minimize the signal propagation delay, i.e., choose routing paths with the lowest delay. To achieve better performance, modifications to the inter-FPGA topology are permitted during the routing process.

4. Problem Description

This problem falls within the domain of digital circuit design, specifically the verification and emulation stages of the Electronic Design Automation (EDA) flow. Given a partitioned logical netlist and a multi-FPGA platform topology, the task is to map all inter-FPGA signal connections onto the available physical interconnects, such that all communication demands are met and the maximum path delay is minimized under resource and delay constraints.

The symbols and notation used in this problem are summarized in Table 1.

Table 1: Notation Summary

Symbol	Description
$G = (V, E)$	An undirected weighted graph, where V is the set of vertices (FPGAs), and E is the set of edges (inter-FPGA links).
$w(F_i, F_j)$	The number of physical communication channels between FPGA F_i and FPGA F_j
n_i	A net in the netlist, representing a group of logical connections between circuit components
s_i	The source node of net n_i
t_{ij}	A sink node of net n_i, connected to source s_i

P_{ij}	The routing path from source s_i to t_{ij}
$D(P_{ij})$	The total delay of path P_{ij}
N_{pq}	The set of nets routed through the physical channel between FPGA F_p and FPGA F_q
r_{pq}	The TDM ratio between FPGA F_p and FPGA F_q, i.e., the number of signals multiplexed on their shared channel
$D(r_{pq})$	The TDM delay associated with TDM ratio r_{pq}
$D(n_i)$	The delay of net n_i, defined as the maximum path delay among all source-to-sink paths within n_i
$\delta_{(F_p,F_q)}$	The change in the number of physical connections between FPGA F_p and FPGA F_q during topology reconfiguration

Due to the limited number of physical interconnect channels between FPGAs, Time-Division Multiplexing (TDM) IP modules are typically deployed at FPGA boundaries to enable high-bandwidth signal transmission across FPGAs. These modules allow multiple logical signals to be time-multiplexed over shared physical links, thereby logically expanding the effective communication bandwidth of the physical channels.

For any given physical channel, if it carries rrr distinct logical signals in a time-multiplexed fashion, its TDM ratio is defined as rrr. In other words, the TDM ratio specifies the number of independent signals that can be transmitted through the channel within one complete TDM cycle. Equivalently, each signal traversing the channel is associated with a TDM ratio equal to that of the channel itself.

In this problem, a uniform TDM ratio is applied to all physical channels connecting any given pair of FPGAs.

Input:

i. A Multi-FPGA Platform Topology

The interconnect structure of the system is modeled as an undirected weighted graph $G = (V, E)$. As illustrated in Figure 1, the vertex set V represents the collection of all FPGAs in the system, while the edge set E corresponds to the physical interconnection channels between FPGA pairs. These inter-FPGA links support Time-Division Multiplexing (TDM) I/O communication.

Each edge $e \in E$ connects a pair of FPGAs, indicating the presence of one or more physical communication links between them. The weight $w(e)$ of edge e is defined as the number of physical channels available between the two connected FPGAs. In other words, the edge weight quantifies the communication capacity between the two FPGAs. For example, as shown in Figure 1, if FPGA F_1 and FPGA F_2 are interconnected by three independent physical channels, the corresponding edge has a weight of $w(F_1, F_2) = 3$.

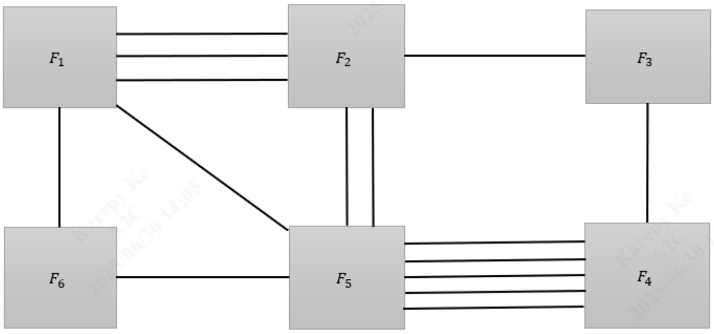

Figure 1 Illustration for a multi-FPGA platform topology

ii. A Partitioned Hypergraph

The logical connectivity of the design is represented as a directed hypergraph $\mathrm{H} = (\mathrm{V}, \mathrm{N})$, where the vertex set V is identical to that used in the topology graph and corresponds to the set of FPGAs. The hyperedge set $\mathrm{N} = \{\mathrm{n}_{1,}\mathrm{n}_2, \ldots, \mathrm{n}_\mathrm{k}\}$ represents the collection of nets, where each net n_i denotes a multi-terminal connection comprising a single source s_i and one or more sink terminals t_ij, all of which are mapped to distinct FPGAs.Formally, each net n_i is a subset of the vertex set V. As illustrated in Figure 2, three representative nets are defined as follows:

- $n_1 = \{s_1, t_{11}, t_{12}\}$
- $n_2 = \{s_2, t_{21}\}$
- $n_3 = \{s_3, t_{31}, t_{32}\}$

Here, s_i denotes the source of communication, and t_ij represents the corresponding sink terminals. All terminals within a net are assumed to be mapped onto different FPGAs.

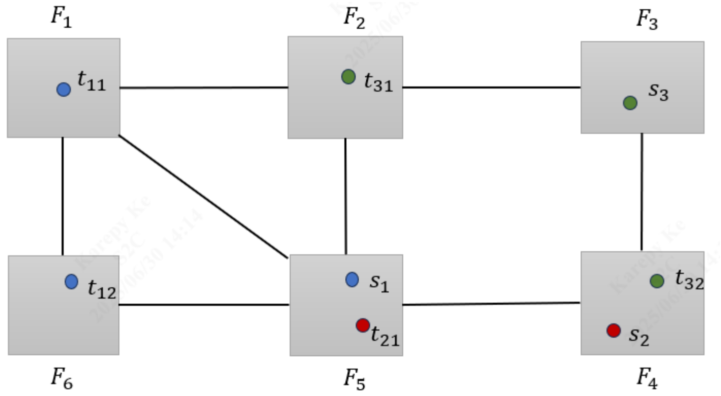

Figure 2 Illustration for the nets terminal mapping.

Output:

For each net n_i, the objective is to assign one or more routing paths in the interconnection graph of the multi-FPGA platform, such that the n_i s_i is successfully connected to all of its sink nodes t_{ij} via paths that may traverse one or more intermediate FPGAs.

Delay Model:

Let P_{ij} denote the routing path from the source node s_i to a sink node t_{ij}. The inter-FPGA delay along this path is denoted by $D(P_{ij})$. This path delay consists of the following two components:

a. TDM Transmission Delay: The delay introduced by time-division multiplexing between every pair of adjacent FPGAs along the path. Specifically, for communication between FPGA F_p and FPGA F_q, the TDM delay is defined as

$$D(r_{pq}) = \alpha * r_{pq} \qquad (1)$$

The TDM ratio r_{pq} between FPGA F_p and FPGA F_q is defined as $r_{pq} = \frac{|N_{pq}|}{w(F_p,F_q)}$, where N_{pq} is the number of nets routed through the physical channel between F_p and F_q. The TDM transmission delay is modeled as proportional to the TDM ratio, with a proportionality constant α (set to $\alpha = 0.7$ in this problem).

b. Interconnect Delay: Each inter-FPGA hop introduces a fixed physical interconnect delay, denoted by the constant β (with $\beta = 30$ in this problem).

Therefore, the total path delay $D(P_{ij})$ is the sum of all inter-FPGA delays along the path, comprising both TDM-induced and fixed interconnect delays.

Net Delay Definition:

The delay of net n_i is defined as the maximum delay among all routing paths from the source to each of its sink nodes:

$$D(n_i) = \max_{1 \le j < |n_i|} D(P_{ij}) \qquad (2)$$

where $|n_i|$ denotes the total number of terminals in net n_i (including the source and sinks).

Optimization Objective:

The objective of this problem is to minimize the worst-case net delay across all nets, formally expressed as:

$$\text{minimize}\left(\max_{1 \le i \le k} D(n_i)\right) \qquad (3)$$

As illustrated in Figure 3, the blue, red, and green paths correspond to the inter-FPGA routing paths assigned to nets n_1, n_2, n_3, respectively.

Take n_1 as an example. It consists of two routing paths from the source node to its sink nodes:

- Path $P_{11}=F_5 \rightarrow F_2 \rightarrow F_1$
- Path $P_{12}=F_5 \rightarrow F_6$

The path P_{11} traverses two FPGA boundaries, and thus its total delay is computed as:

$$D(P_{11}) = D(r_{52}) + D(r_{21}) + 2 * \beta \qquad (4)$$

where:

- $D(r_{pq})=\alpha * r_{pq}$ represents the TDM delay between FPGA F_p and F_q, with r_{pq} denoting the TDM ratio on that link;
- β is the fixed inter-FPGA routing delay incurred at each hop between FPGAs.

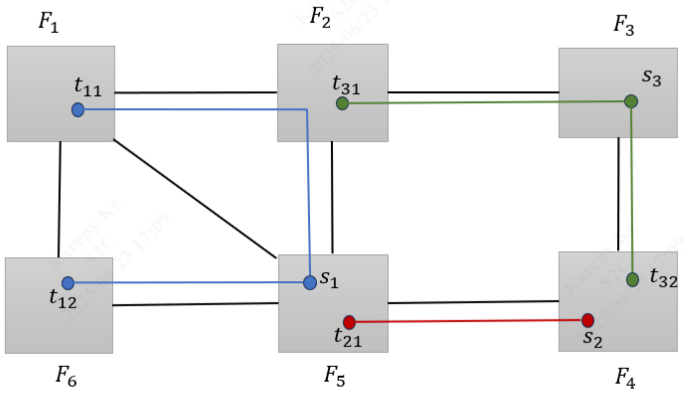

Figure 3 Illustration for a routing result.

The following example illustrates the calculation of the TDM ratio and the counting rule for nets traversing inter-FPGA channels.

As shown in Figure 4, consider the routing configuration of nets n_1, n_2, n_3, n_4:

- The number of physical channels between FPGA pairs is given by:

$$w(F_3, F_4)=2, w(F_4, F_5)=2, w(F_5, F_6)=1 \qquad (5)$$

- The number of nets traversing these channels is:

$$|N_{34}|=2, |N_{45}|=4, |N_{56}|=2 \qquad (6)$$

Therefore, the TDM ratios between the three pairs of FPGAs are respectively:

$$r_{34} = \frac{2}{2} = 1, r_{45} = \frac{4}{2} = 2, r_{56} = \frac{2}{1} = 2 \qquad (7)$$

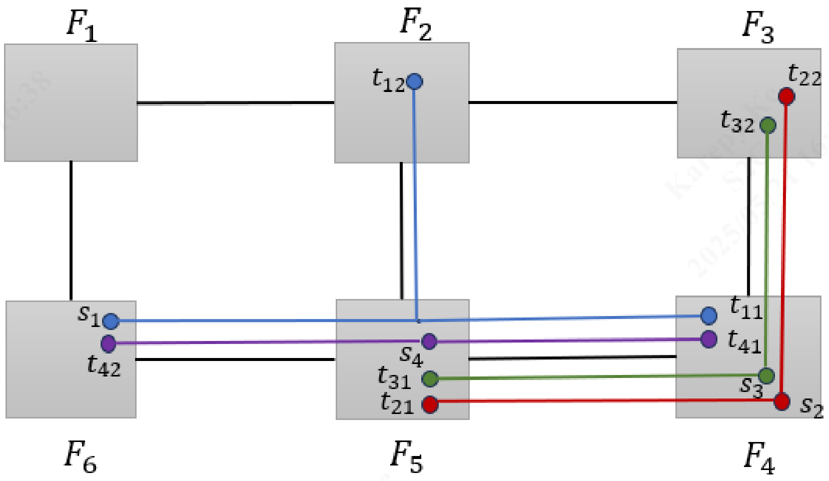

Figure 4 Illustration for the calculation of the TDM ratio.

Net Traversal Counting Rule for Inter-FPGA Channels:

It is important to note that when calculating the number of nets traversing a given inter-FPGA channel, each net is counted only once per channel, even if it passes through the same channel multiple times. For instance, as illustrated in Figure 4, the blue net n_1 contains two paths from the source to its respective sink nodes:

- $P_{11} = F_6 \rightarrow F_5 \rightarrow F_4$
- $P_{12} = F_6 \rightarrow F_5 \rightarrow F_2$

Although both paths traverse the channel $F_6 \rightarrow F_5$, net n_1 is counted only once in the computation of the net set $|N_{65}|$. This rule prevents overestimation of channel utilization due to redundant counting of the same net.

Topology Reconfiguration and Path Delay Optimization:

To further optimize routing delay, dynamic reconfiguration of the inter-FPGA physical topology may be considered during the routing process. This involves adding, removing, or reassigning physical channels between arbitrary pairs of FPGAs to reduce the delay of critical paths.

Example of Topology Adjustment and Delay Reduction:

As shown in Figure 5, assume that initially every pair of adjacent FPGAs is connected by a single physical channel. Under this topology, the maximum path delay arises from net n_1 along path is $P_{11} = F_5 \rightarrow F_2 \rightarrow F_1$, the delay is:

$$D(P_{11}) = D(r_{52}) + D(r_{21}) + 2 * \beta \qquad (8)$$

Observing the topology, it is evident that the channel between FPGAs F_5 and F_2 is exclusively utilized by net n_1. If this channel is reassigned to connect F_5 and F_1, the connectivity of other nets (e.g., n_2, n_3) remains unaffected.

Consequently, the routing path P_{11} can be updated to a shorter path:

$$P_{11} = F_5 \rightarrow F_1, D(P_{11}) = D(r_{51}) + \beta \qquad (9)$$

This reconfiguration significantly reduces the maximum path delay, thereby improving the overall system performance.

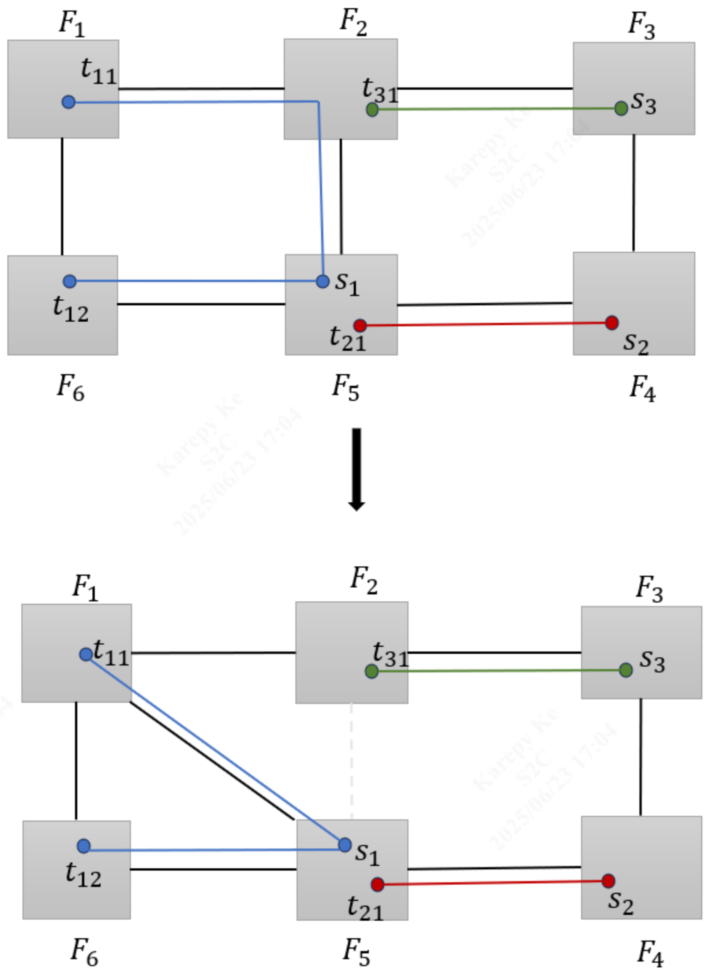

Figure 5 Illustration for a reconfiguration.

Explanation of Channel Modification Metrics:

To quantitatively characterize the impact of network reconfiguration on physical resources, we define the concept of channel modification quantity. Let $\delta_{(\mathrm{F_p},\mathrm{F_q})}$ denote the modification in the number of physical interconnect channels between FPGAs $\mathrm{F_p}$ and $\mathrm{F_q}$. The interpretation of $\delta_{(\mathrm{F_p},\mathrm{F_q})}$ is as follows:

- $\delta_{(F_p,F_q)} > 0$: indicates that new physical channels are added between F_p and F_q;
- $\delta_{(F_p,F_q)} < 0$: indicates that existing physical channels are removed between F_p and F_q;

- $\delta_{(F_p,F_q)} = 0$: indicates that the physical channel configuration remains unchanged.

In the aforementioned example, the reconfiguration results in:

$$\delta_{(F_5,F_1)} = +1, \delta_{(F_5,F_2)} = -1 \qquad (10)$$

All other FPGA pairs exhibit no change, i.e., $\delta_{(F_p,F_q)} = 0$.

Hence, the total number of modified inter-FPGA channels in this reconfiguration is:

$$\sum_{(F_p,F_q)} \left|\delta_{(F_p,F_q)}\right| = 2 \qquad (11)$$

5. Evaluation Criteria

During the process of physical network reconfiguration and routing optimization, the following constraints must be strictly satisfied:

a. TDM Ratio Constraint

For any connected pair of FPGAs F_p and F_q, the time-division multiplexing (TDM) ratio r_{pq} must not exceed a system-defined maximum value R_{max}. That is:

$$r_{pq} \leq R_{max}, \forall (F_p, F_q) \in E \qquad (12)$$

b. TDM Ratio Alignment Constraint

All TDM ratios must be integer multiples of 8 to comply with the hardware requirements of the TDM IP module:

$$r_{pq} \in \{8, 16, 24, \dots\}, \forall (F_p, F_q) \in E \qquad (13)$$

c. FPGA Channel Capacity Constraint

For each FPGA F_p, the total number of its outgoing physical interconnect channels must not exceed a predefined maximum $W_{max}(F_p)$. The total number of outgoing channels for F_p is given by:

$$\sum_{F_q \in V, p \neq q} w(F_p, F_q) \leq W_{max}(F_p), \forall F_p \in V \qquad (14)$$

d. Channel Reconfiguration Scope Constraint

The total absolute change in the number of physical interconnect channels across the network, i.e., the sum of the absolute values of all channel modifications $\delta_{(F_p, F_q)}$:

$$\sum_{(F_p, F_q) \in (V \times V)} |\delta_{(F_p, F_q)}| \qquad (15)$$

must not exceed 30% of the total number of physical channels in the current topology. Let W_{total}denote the total number of channels in the initial configuration. Then the constraint is:

$$\sum_{(F_p, F_q) \in (V \times V)} |\delta_{(F_p, F_q)}| \leq 0.3 \cdot W_{total} \qquad (16)$$

Example: In the case illustrated in Figure 5, the total number of physical channels is 7. After reconfiguration, the total modification is 2, which satisfies this constraint.

Runtime Limit:

To prevent overloading the evaluation server, this competition imposes a strict runtime limit. For all test cases, the total execution time—including input parsing and result output—must not exceed **1 hour**. While correctness is the primary criterion, runtime performance is also a factor in the overall evaluation. **Multithreading is not supported** in this competition.

Memory Limit:

Throughout all test cases, the peak memory usage must remain under **32 GB**.

Scoring Criteria:

For each test case, the organizing committee will evaluate the executable submitted by participants using a designated assessment tool. The results will be scored and ranked accordingly. Suppose there are **20 participating teams**: the team with the best performance receives **20 points**, the second-best receives **19 points**, and so on. The final score will be a **weighted sum of scores across all test cases**.

For each individual test case, a **lower score indicates better performance**, and the score is computed as:

$$score = x * \left(1 + 0.2 * \frac{y}{3600}\right) \qquad (17)$$

Where:

- x is the **maximum path delay** in the routing result
- y is the execution time in seconds

If a submitted solution is invalid or its execution exceeds the 1-hour limit, it will be **ranked last** for that test case.

Participants are required to submit an executable file that complies with the aforementioned specifications. If dynamic libraries are used, all necessary runtime dependencies must be included. It is the participant’s responsibility to ensure that the submitted executable runs correctly on the competition server.

6. Benchmark Description

The number of pins in the benchmark is defined as the total number of nodes across all hyperedges, formally expressed as:

$$Number\ of\ pins = \sum_{e \in E} |e| \qquad (18)$$

The scale of all benchmark instances is summarized in the table below. Note that the scoring weights assigned to public benchmarks are lower than those assigned to hidden benchmarks.

Table 2. Overview of Benchmark Instances

ID	Visibility	Weight	Data Scale
1	public	0.6	Number of nodes ≤ 100, number of FPGAs ≤ 4
2			Number of nodes ≤ 10,000, number of FPGAs ≤ 8
3			Number of nodes ≤ 200,000, number of FPGAs ≤ 32
4			Number of nodes ≤ 5,000,000, number of FPGAs ≤ 64
5-6	private	1.0	Number of nodes ≤ 10,000, number of FPGAs ≤ 8
7-8			Number of nodes ≤ 200,000, number of FPGAs ≤ 32
9			Number of nodes ≤ 1,000,000, number of FPGAs ≤ 64
10			Number of nodes ≤ 5,000,000, number of FPGAs ≤ 64

7. Input and Output File Specifications

Input Files:

- **design.info**:

  Specifies the resource constraints of each FPGA node in the system, particularly the maximum number of allowable outbound physical connection channels for each FPGA.

- **design.net**:

  Describes the communication network of the system. Each line represents a hyperedge (net), containing the net's weight (e.g., communication intensity or bandwidth requirement) and its associated logical nodes (including the source and multiple sinks), forming a directed hypergraph.

- **design.topo**:

  Specifies the physical interconnection topology among FPGAs. This file provides the connection channel information between each pair of FPGAs, thereby defining the initial topology graph $G=(V, E)$.

- **design.fpga.out**:

  Defines the mapping results from logical nodes to physical FPGAs, i.e., the partitioning outcome of the hypergraph. This file indicates the FPGA assignment for each logical node and serves as the positional basis for subsequent routing.

Output Files:

- **design.route.out**:

  Represents the output of the routing phase, detailing the specific routing paths between the source and all sink nodes for each net. Each path is represented as an ordered sequence of FPGA nodes traversed, reflecting the actual signal transmission route across the multi-FPGA system.

- **design.newtopo**:

  Represents the reconfigured physical interconnection topology after network restructuring. This file captures the modifications (additions, deletions, or alterations) applied to the initial topology during optimization and records the updated number of connection channels between each pair of FPGA nodes in the new topology graph.

8. Example

Input Example (Figure 6)

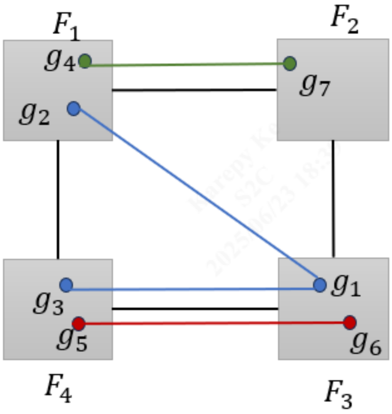

Figure 6 Input example.

design.info — Format Specification (Resource Constraints):

This file specifies the physical connection resource constraints for each FPGA node in the system. The file is organized line by line, where each line corresponds to an individual FPGA node, following the format:

<FPGA_ID> <Max_IO>

- <FPGA_ID>： A unique identifier for the FPGA (e.g., F1, F2, etc.).
- <Max_IO>： The maximum number of outbound physical communication channels allowed for this FPGA, indicating the upper limit of its available I/O resources.

Example:

The following entries indicate that there are four FPGAs in the system, and each has a maximum of 3 external communication links:

F1 3

F2 3

F3 3

F4 3

design.net — Format Specification:

This file describes the communication requirements of the system, represented as a hypergraph capturing the logical signal connections. The file is organized line by line, where each line corresponds to a net (hyperedge) and follows the format:

<Source_Node> <Weight> <Sink_Node_1> <Sink_Node_2> ...

- <Source_Node>：The source terminal of the net.
- <Weight>：The weight of the net, representing the communication intensity or priority. In this competition, all nets are assigned a uniform weight of 1.
- <Sink_Node_i>：A sink terminal of the net. Each line may contain multiple sink nodes.

For example, the following entries define three nets:

g1 1 g2 g3

g4 1 g7

g5 1 g6

design.topo — Format Specification (Initial Physical Topology):

This file specifies the initial physical interconnection topology among all FPGA nodes in the system. The connections are encoded using an adjacency matrix format, where each entry indicates the number of physical communication channels between two FPGA nodes. The file is organized line by line, with each line describing the connectivity of one FPGA node. The format of each line is:

<FPGA_ID>: <C_1>, <C_2>, ..., <C_n>

- <FPGA_ID>：The identifier of the FPGA node being described
- <C_i>：The number of physical communication channels between the current FPGA and the *i-th* FPGA node (according to a globally consistent index).
- The entries are ordered by FPGA indices, and include self-connection (i.e., the diagonal element, typically zero).

This format is functionally equivalent to an $n \times n$ symmetric adjacency matrix (assuming bidirectional links), where the *i-th* row encodes the number of channels between FPGA *F_i* and every other FPGA node (including itself).

Example (corresponding to Figure 6):

F1: 0,1,0,1

F2: 1,0,1,0

F3: 0,1,0,1

F4: 1,0,1,0

This matrix describes a system of four FPGAs, where, for example, FPGA F1 is connected to F2 and F4 via one physical channel each.

design.fpga.out-- Format Specification :

This file records the netlist partitioning result, mapping logical nodes to their assigned FPGA devices. Each line corresponds to one FPGA and lists the logical nodes allocated to it as a result of the hypergraph partitioning process. The format is as follows:

<FPGA_ID>: <node_1> <node_2> ... <node_3>

Example (corresponding to Figure 6):

F1: g2 g4

F2: g7

F3: g1 g6

F4: g3 g5

In this example, nodes g2 and g4 are assigned to FPGA F1, node g7 to F2, and so on.

Output Example (Figure 7)

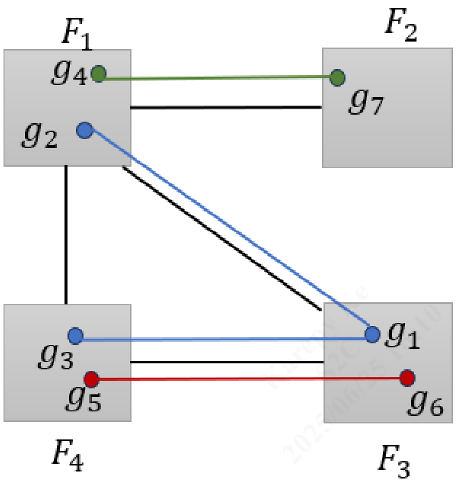

Figure 7 Output example.

design.route.out (Routing Result File):

This file records the detailed routing path information for each logical net. The nets are sorted in descending order based on their maximum path delay. Each net occupies several lines in the following format:

[Net ID]

[FPGA sequence to Sink 1] [path delay]

[FPGA sequence to Sink 2] [path delay]

…

- [Net ID]：The identifier of the logical net (starting from 1), corresponding to the line number in design.net.
- [FPGA sequence to Sink]：The routing path from the net's source FPGA to one of its sink FPGAs, represented as a sequence of FPGA index numbers in traversal order.
- [path delay]：The cumulative delay of the path, calculated as the sum of routing and TDM delays across all inter-FPGA hops.

Example (Figure 7):

[net 1]

[3,4] [35.6]

[3,1] [35.6]

This example indicates that Net 1 (corresponding to the blue net in Figure 7) has two sink terminals:

- The path from source FPGA F3 (index 3) to sink FPGA F4 (index 4) is: F3 → F4, with a path delay of 30 + 0.7 × 8 = **35.6**.
- The path to sink FPGA F1 (index 1) is: F3 → F1, with the same delay of **35.6**.

All nets are sorted in descending order by their maximum path delay $D(n_i) = \max_j D(P_{ij})$ to highlight the critical paths in the routing results.

design.newtopo (Post-Optimization Physical Topology):

This file describes the updated physical interconnection topology among FPGA nodes after routing optimization. The format mirrors that of the input file design.topo, utilizing an adjacency matrix to represent the number of physical channels between FPGA pairs.

Each line corresponds to a single FPGA node and follows the format:

<FPGA_ID>: <C_1>, <C_2>, ..., <C_n>

Example (Figure 7):

F1: 0,1,1,1

F2: 1,0,0,0

F3: 1,0,0,1

F4: 1,0,1,0

9. References

[1] Lin T W, Tai W C, Lin Y C, et al. Routing topology and time-division multiplexing co-optimization for multi-FPGA systems[C]//2020 57th ACM/IEEE Design Automation Conference (DAC). IEEE, 2020: 1-6.

[2] Zou P, Lin Z, Shi X, et al. Time-division multiplexing based system-level FPGA routing for logic verification[C]//2020 57th ACM/IEEE Design Automation Conference (DAC). IEEE, 2020: 1-6.

[3] Turki M, Marrakchi Z, Mehrez H, et al. Iterative routing algorithm of inter-FPGA signals for multi-FPGA prototyping platform[C]//International Symposium on Applied Reconfigurable Computing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013: 210-217.

[4] Khalid M A S, Rose J. A novel and efficient routing architecture for multi-FPGA systems[J]. IEEE transactions on very large scale integration (VLSI) systems, 2000, 8(1): 30-39.

[5] McMurchie L, Ebeling C. PathFinder: A negotiation-based performance-driven router for FPGAs[C]//Proceedings of the 1995 ACM third international symposium on Field-programmable gate arrays. 1995: 111-117.

[6] Huang, K., Wu, Z., & Wang, W. *Very Large Scale Integrated Circuit Physical Design: From Graph Partitioning to Timing Closure*. Beijing: Publishing House of Electronics Industry, 2021. (in Chinese) .

*For questions not covered in this guide, please refer to the Q&A document