

第五届（2023）集成电路 EDA 设计精英挑战赛 赛题指南

Document Change Log

Version 1.00	
20231011	新增六、评分细则。
20230918	初始版本

一、赛题：

FPGA Die 级系统布线算法设计

二、命题企业：

思尔芯

三、赛题背景：

FPGA 内部常常由几颗 SLR（Super Logic Region）组成，也称为 Die，Die 之间通过 SLL（Super Long Line）按照一定的连接关系进行连接。同一颗 FPGA 中，具有 SLL 连接的两个 Die 之间的“空隙”被称为 Die Crossing。

目前，在大规模集成电路的原型验证过程中，需将设计电路分割至多颗 FPGA 中后，再进行 Die 级更细粒度的分割，并在 Die 级分割结果的基础上进行 Die 之间信号的布线，也即 Die Level System Routing。

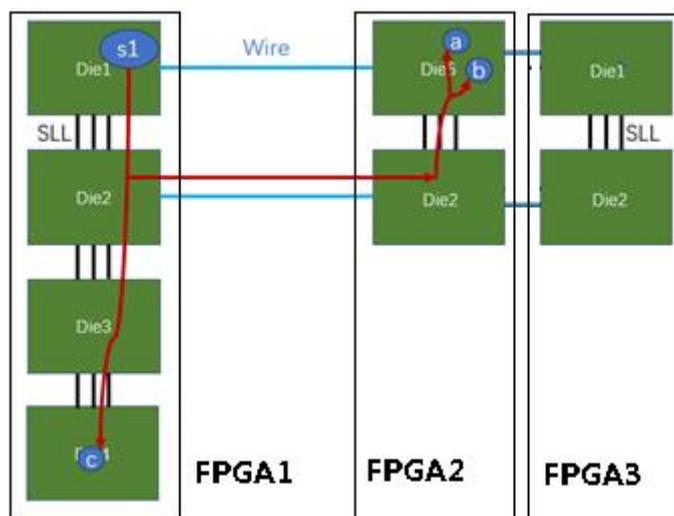


图 1 Die Routing 示意图

在 Die 级布线的一些特殊约束下，发现 Die Crossing 处的 SLL

消耗过量,有些情况下甚至不能完成布线,同时信号的延时时间过长,影响系统运行频率。

因此,需求一种 **Die** 级布线算法,在考虑组网、**TDM** 等相关约束条件下,围绕给定的优化目标,输出符合格式要求的布线结果。该赛题着重考察了学生的算法设计能力,EDA 相关背景知识,程序设计与编写、调试等能力。不限于计算机、数学、电子等等专业的同学参加。

四、赛题描述:

FPGA 原型验证是一种使用现场可编程门阵列 (**FPGA**) 芯片来验证硬件设计的方法。在 **FPGA** 原型验证中,硬件设计师将设计编程到 **FPGA** 芯片中,通过将设计映射到 **FPGA** 上并连接外部接口和其他组件,创建一个可以运行的硬件原型。这样的原型可以更接近最终的硬件实现,使设计团队能够在实际硬件环境中进行验证和测试。

原型验证的组网就是逻辑阵列组建的硬件网络,具体包括逻辑阵列型号和资源总量、逻辑阵列数量、拓扑结构、互连线数量和连接关系等等。本赛题中,组网信息将会细化到 **Die** 级的颗粒度。

如上文所述,本赛题在 **Die** 级分割结果的基础上进行 **Die** 之间信号的布线,也就是说每个 **Die** 上的电路逻辑已经固定,在赛题中各个电路逻辑用带序号的不同节点表示。

由于 **FPGA** 之间物理连线(**Wire**)数量的限制,通常需要在 **FPGA**

边界上通过插入 TDM（时分复用）IP 模块，完成信号传输的时分复用，逻辑上动态扩展了物理连线的数量。对某个 Wire 来说，以时分复用方式通过它的信号的数量称为该 Wire 的 TDM Ratio，同时我们也常常这样表述：经过该 Wire 的每个信号的 TDM Ratio 就等于该 Wire 的 TDM Ratio。同一个 Wire 中时分复用的信号传递方向必须相同，不同 Wire 中信号的传递方向可以不同，不同 Wire 的 TDM Ratio 可以不同。信号在经过某对 FPGA 之间的 TDM Ratio 越大，那么该信号的延时越大，信号经过某对 FPGA 之间的 TDM Ratio 的大小与对该信号产生延时可以用特定公式来计算。且 TDM Ratio 的取值必须是某个基数的整倍数，这是由 TDM IP 决定的，例如在基数为 4 的 TDM IP 中，如果 3 个信号共享 1 个 wire，那么这 3 个信号的 TDM Ratio 要向上取整，最终为 Ratio=4，该 TDM 要补一个空信号。

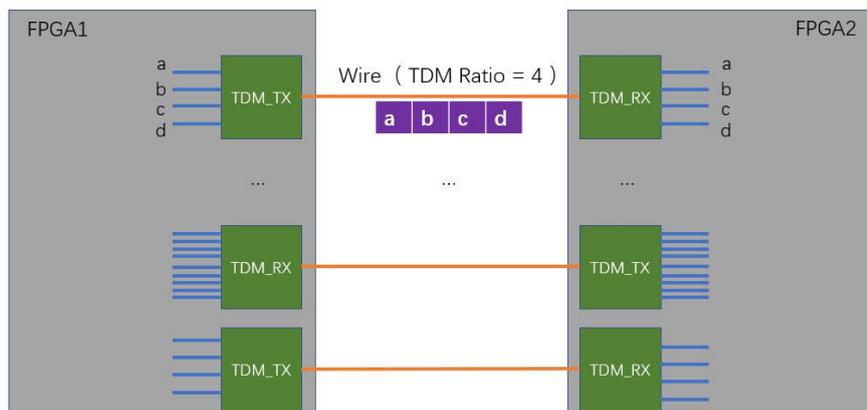


图 2 TDM 原理示意图

FPGA 之间的信号传递可以使用 TDM，在 FPGA 内部传递的信号不可以使用 TDM，也就是说信号在跨越 FPGA 的时候，可以使用 TDM 扩展逻辑上的物理连线数量，即信号数量可以大于 FPGA 之间

物理连线数量；但 Die Crossing 之间的 SLL 数量却是绝对的，即跨越 Die 的信号数量必须小于或者等于 SLL，信号数量不允许超过 SLL 数量，这也是 Die Level Routing 的一个重要约束。本赛题中经由 SLR 的信号延时 Delay 取值为 1，TDM Delay 计算公式统一为 $Delay = A + B * Ratio$ ，Ratio 基数为 4，向上按倍数逐步增加，A 为常量 1，B 为常量 2。如图 2 中，信号 a, b, c, d 在经过 FPGA1 和 FPGA2 之间的 TDM 时，获得的 Delay 值均为 $1 + 2 * 4 = 9$ 。

Die 级布线的优化目标为最小化 net 的最大延时。net 也指本文中提及的“信号”，一个 net 只有一个驱动节点，可以有超过两个的负载节点，如图，该 net 具有 1 个驱动节点 s，同时驱动 3 个负载节点 a, b, c，如上文所述节点代表电路模块。

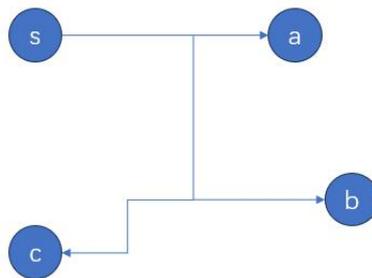


图 3 Net 示意图

其中，net 相连的各个节点可能在分割过程中分配至不同的 Die 中，即图中的 s, a, b, c 节点可能所处组网中任何位置，不妨假设如图 1 所示，s 节点被分割至 FPGA1 的 Die1 中，a, b 节点被分割

至 FPGA2 的 Die1, c 节点被分割至 FPGA1 Die4。Routing 的目的就是将 net 的所有节点 a, b, c, d 在组网中连接起来并使得目标值最小, 其中 net 在 FPGA 内部 Die 之间和在跨过 FPGA 时产生延时。

赛题要求在指定的组网和上述约束条件内, 完成给定所有 net 的 Routing 处理, 在满足跨越 Die 的信号数量必须小于或者等于 SLL 条件下, 使得所有 net 目标值 RoutingWeight 最大值最小。

单个 net 对应的 *Routing Weight* 值函数 $f(\text{net}) = \text{Max}(\sum(\text{Tc} * \text{Delay}))$, 其中 Tc 为该 net 相关的时序系数, **同个 FPGA 内的 Die 之间的 Tc 取值为横跨 SLR 的个数, 不同 FPGA 间的 Tc 取值为 0.5。** Delay 为该 net 驱动节点到每个负载节点间的路由路径的延时值。

举例说明, 一条 net 的 Routing 路径延时如下图所示, 图中例如 D1(=2)为信号在 FPGA1 内部跨 Die 产生的延时, D2(=9)为信号经由 FPGA1 到 FPGA3 通过 TDM 产生的延时, D3(=9)为信号经由 FPGA3 到 FPGA4 通过 TDM 产生的延时, D4(=17)为信号经由 FPGA4 到 FPGA5 通过 TDM 产生的延时, D5(=1)为信号经由 FPGA4 内部 1 个 Die 产生的延时。该 net 驱动节点 s2 分别到达 e, f, g 三个负载的路由路径延时为 D1, RW=2 (s2->e), D2+D3+D5+D4, RW=9*0.5+9*0.5+1+17*0.5=18.5(s2->f) , D2=9,RW=9*0.5=4.5 (s2->g), 三者中的最大值 18.5 为该 net 的 *Routing Weight*。

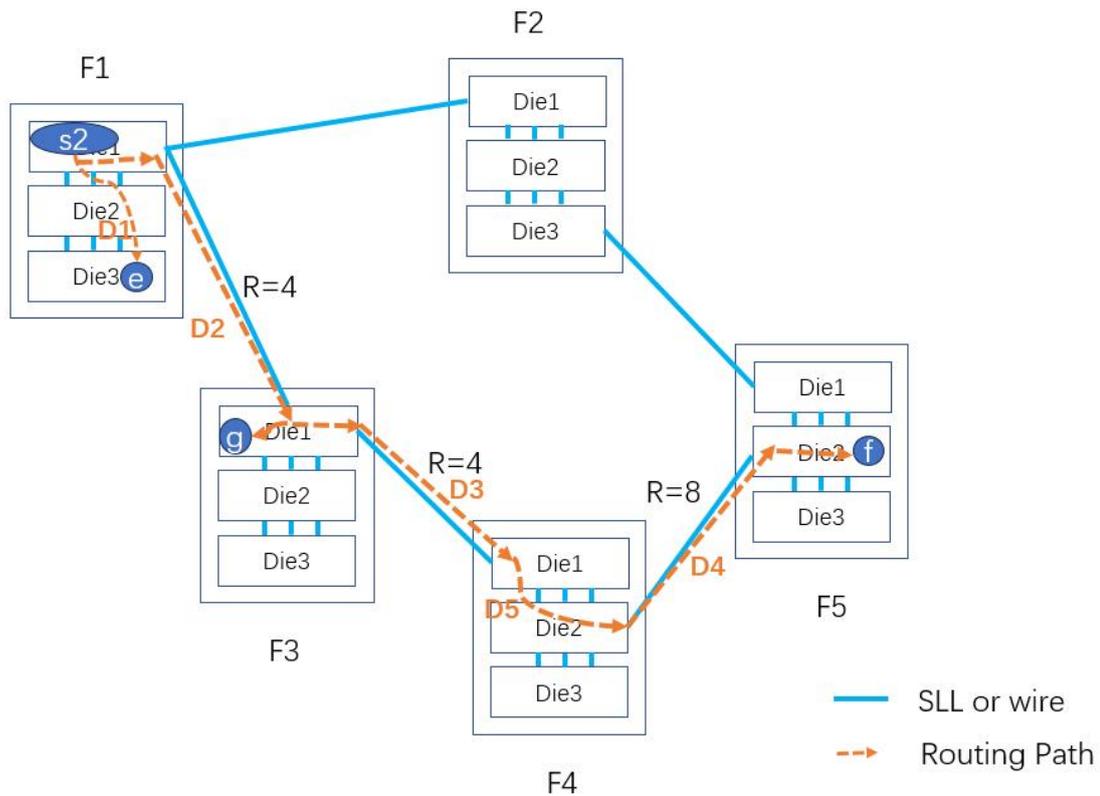


图 4 Routing Weight 计算示意图

涉及知识点:

- 布线算法

五、赛题接口文件

1.1 输入文件

- Net Definition File <连线定义文件 design.net>

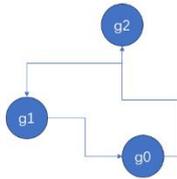
Net 的描述文件，每行表示一个 net 的驱动节点或者负载节点信息，格式如下，
 <node> <s/l> [weight]

s 含有驱动(driver)节点的连线部分
 l 含有负载(load)节点的连线部分，一个连线可能含有一个或多个负载部分
 weight 连线的权重值,可选,本赛题都按照 weight = 1 来处理。
 任意 net 的 s 节点在文件中所在的行号，为该 net 的编号，用于彼此区分。

例如，

```

g1 s 1
g0 l
g0 s 1
g2 l
g1 l
  
```



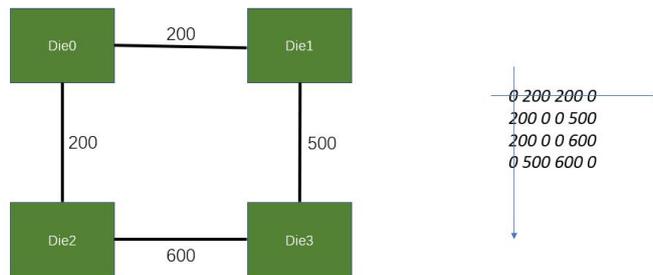
- **Die Level Network File <Die 级组网描述文件 design.die.network>**

Die 级的组网描述文件：假设组网中共存在 N 个 Die，每个 Die 具有唯一编号，该编号从 0 开始。组网为一个 $N \times N$ 的矩阵，用来描述各 Die 之间的连接关系和连线的数量，这里的连线有可能是 SLL，也有可能是 FPGA 之间的 Wire，类别取决于两个 Die 是否处于同一个 FPGA，该信息包含在 design.fpga.die 中，在 design.die.network 中无法知道哪些 die 属于同一个 FPGA。

格式例如，矩阵为对称矩阵，其中的每个位置的值为对应横纵坐标 Die 编号之间的连线数量。

```

0 200 200 0
200 0 0 500
200 0 0 600
0 500 600 0
  
```



- FPGA Die Relation File <FPGA 与 Die 的关系文件

`design.fpga.die`

描述组网中所有 FPGA 和 Die 之间归属关系的文件，即描述哪些 Die 属于同一颗 FPGA。

格式为：

`[fpga]:[die]`

举例：

FPGA0: Die0 Die2

FPGA1: Die1 Die3

表示 Die0 Die2 属于同一颗 FPGA，该 FPGA 为 FPGA0。

表示 Die1 Die3 属于同一颗 FPGA，该 FPGA 为 FPGA1。

- Node Die Position File <节点 Die 级分割位置文件

`design.die.position`

节点所在 Die 的位置文件，该文件每一行包含一个分组信息里面包含的节点列表，

以“Die 序号:”开始，每个节点之间以空格隔开，可以多行表示。

格式如下，

`<Die Index>: <node-list>`

例如，

`Die0:g0 g1 g19
g20 g21 gp0`

`Die1:g100 gp1 gp2`

节点名称有些以 g 开头，有些以 gp 开头，g 节点和 gp 节点只是名称不同。

1.2 输出文件

- **The Routing Result File<布线结果文件 design.route.out>**

该文件应包含每条 routing net 的 Net Max Routing Weight, 以及对应的 net 编号, 以及该 net 的布线路径。Net 先后顺序按照 Net Max Routing Weight 从大到小的顺序排列, 格式为:

```
[Net ID]
[Die Index on Driver to Loader1 Routing Path][Net Routing Weight]
[Die Index on Driver to Loader2 Routing Path][Net Routing Weight]
...
```

例如

```
[3]
[1,2,5,6][50]
[1,2,3][48]
```

表示 id 为 3 的 net, 从 driver 到达第一个 loader 的布线路径为 Die1,2,5,6, routing weight 为 50; 从 driver 到达第二个 loader 的布线路径为 Die1,2,3, routing weight 为 48。

路径包括 net 驱动节点和负载节点所在的 Die 以及布线路径经过的 Die。

- **The TDM Result File<TDM 分配结果文件 design.tdm.out>**

该文件应包含具有 Wire 连接的所有 Die 之间, 经过信号的时分复用组合结果, 具体应包含所有具有 Wire 连接的 Die Index 之间, 共享一个 Wire 的 Net ID 以及该 Wire 的 TDM Ratio。格式为:

```
[Die Index1] [Die Index2]
[Net id...] TDM Ratio //每行代表一个 Wire
...
```

例如

```
[Die2,Die3]
[3,5,6,7] 4
[15,9,0,8,11,12,13,20] 8
```

表示 Die2 和 Die3 之间:

第一个 wire, 时分复用的 net id 组合为 net 3,5,6,7, 它的 TDM Ratio 为 4;
第二个 wire, 时分复用的 net id 组合为 net 15,9,0,8,11,12,13,20 它的 TDM Ratio 为 8;

...

六、评分标准:

技术评分点:

- 在满足赛题约束的情况下得到正确布线结果
- 满足正确性的前提下比较优化目标
- 程序的运行时间和内存使用大小
- 提供按照要求的算法设计方案报告和程序操作说明等

评分原则重要性依次降低。

技术评分细则，共 100 分。

1. 基础任务 - 60 分（Benchmark 测试的完成度占比 70%，结果正确性、最优化、运行速度、内存使用量指标占比 30%）；

测试项	测试描述
testdata-1	8 个 Die, 2 颗 FPGA, 节点数目 (6)
testdata-2	8 个 Die, 2 颗 FPGA, 节点数目 (71)
testdata-3	8 个 Die, 2 颗 FPGA, 节点数目 (69)
testdata-4	8 个 Die, 2 颗 FPGA, 节点数目 (452)
testdata-5	12 个 Die, 3 颗 FPGA, 节点数目 (5084)
testdata-6	12 个 Die, 3 颗 FPGA, 节点数目 (120000)

2. 高性能任务 - 40 分。（Benchmark 测试的完成度占比 70%，结果正确性、最优化、运行速度、内存使用量指标占比 30%）。

测试项	测试描述
testdata-7	16 个 Die, 4 颗 FPGA, 节点数目 (50000)
testdata-8	16 个 Die, 4 颗 FPGA, 节点数目 (60000)
testdata-9	16 个 Die, 4 颗 FPGA, 节点数目 (700000)
testdata-10	20 个 Die, 5 颗 FPGA, 节点数目 (3000000)