

第四届 (2022) 集成电路 EDA 设计精英挑战赛

赛题指南

一、**赛题名称:** 动态组网算法设计 (数字集成电路设计方向)

二、**命题企业:** 上海思尔芯技术股份有限公司

三、**赛题 Chair:** 余立艳 西安电子科技大学、叶佐昌 清华大学

四、**赛题背景:**

随着电路设计的复杂性不断增加,在逻辑仿真和快速原型设计中电路设计需划分成多份,并分配到多 FPGA 系统进行仿真和验证。多 FPGA 系统是人工预先设计系统组网和连接关系,在此约束下对电路进行划分处理。由于人工预设的组网约束不合理,往往造成划分失败,划分质量不佳等问题。现需求一种新算法,支持动态组网,满足划分约束,输出最优划分结果和组网方案。本赛题为 2020 年赛题的延续增强版,侧重解决工程实施问题和质量提升。

五、**赛题描述:**

动态组网

组网就是逻辑阵列 (FPGA) 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。组网约束指硬件限制条件的集合,如互连线连接关系和数量限制、逻辑阵列资源限制等等。

划分过程是在特定组网,即在特定硬件网络上进行的划分,也就是说,是在特定组网约束条件下的划分,划分结果必须满足组网约束条件,同时适合用户设计结构的组网结构会极大地提高划分后系统的性能。所以,组网和设计划分是相辅相成、互相作用、互相配合的关系:组网是划分的硬件条件限制,即划分结果

如果不能被硬件条件所支持,如划分结果所需要的硬件资源数量,硬件系统不能支持,那么此划分结果是无用的;划分是组网的服务目标,组网可以根据设计特点调整,合理、高效的组网能够帮助划分结果提高运行性能,如逻辑阵列资源利用合理充分、高效,信号就会传递快速、流畅等。

当前,业内通常凭借经验,根据用户设计特征,人工设计组网结构后,再进行用户设计的划分和 FPGA 位置分配,组网约束可能不合理,可能造成划分失败。本赛题要求的动态组网,期望划分算法在平面网格内和一些组网约束的前提下,能够结合用户设计,通过程序科学规划组网结构,包括所使用 FPGA 的数量、拓扑结构、连接关系等。使该组网能够高效配合划分过程,得到最优化的划分结果。

动态组网过程将在以下输入约束的情况下进行:

1、 划分的基本硬件单元可配置为 1/2/4 颗 FPGA,也即对应了 single/dual/quad 类型的原型板。划分的基本单元是指由 FPGA 按照一定形状组合成的硬件单元,有些单元只包含一颗 FPGA,有些则可能包含两颗或者四颗,如图 1 所示。Single 类型的划分基本单元只包含一颗 FPGA,组网比较灵活; Dual 类型与 Quad 类型分别由两颗和四颗 FPGA 以固定的拓扑结构结合而成。划分基本单元要求摆放在平面网格中,网格的最小正方形边长计为单位 1。

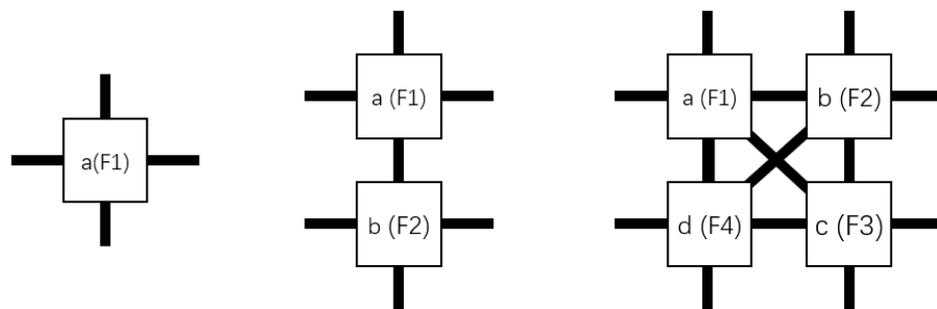


图 1 分割基本单元结构示意图

相比于单颗 FPGA 组成的划分单元,其他类型划分单元在组网中相对位置变化的情况更多,更为复杂,如图 2 所示。图中以 Dual 类型 FPGA 为例,不同颜色的长方形代表着不同的 Dual 单元,四个 Dual 单元被摆放至不同的位置,其中两颗 FPGA 的距离按照平面距离计算,如图 2 中 a1 与 a2 的距离为 $\sqrt{5}$ 个单位, a2 与 b2 的距离为 1 个单位,依此类推。不同种类的硬件单元具有不同的资源数量和对外连接端口数量。在某次划分过程中将指定使用唯一一种划分单元类型组网,不存在混合类型的情况。

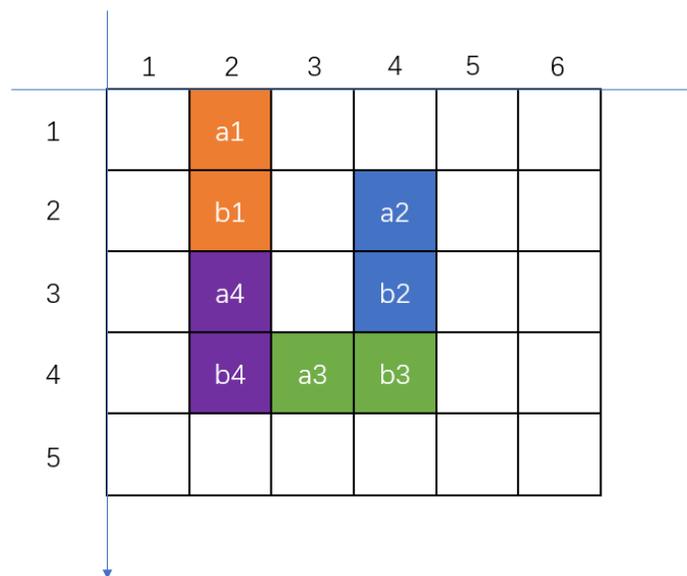
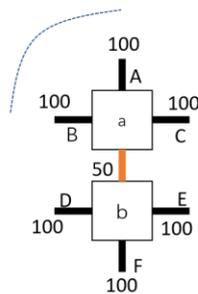


图 2 分割单元相对位置示意图

分割基本单元的对外连接线总数和内部连接线总数将会按照连接点进行均分,如在图 3 中, Dual 类型的基本分割单元,对外连接总数假如为 600,则其 6 个使用的对外连接点 (ABCDEF) 的互连线数量各为 100;如果内部互连线数量为 50,则内部连线为 50。同理,对于 Quad 类型的基本分割单元,假设其对外连接总数为 1600,则每个对外连接点 (ABCDEFGH) 的对外连线总数为 1600 除以 8,为 200;同理如果其内部互连线总数为 600,则其内部连线 IJKLMN,

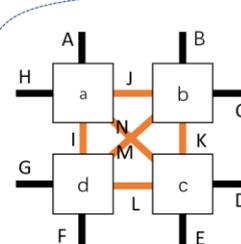
每根分得 100 根互连线。Single 类型基本分割单元则无内部连接线，其对外 4 个使用的连接点同样均分外部连接线。unit.max_int 规定一颗 FPGA 最多与 unit.max_int 颗其他 FPGA 通过连接线进行直接连接，其中最大连接数量也是按照均分的原则分配，如图中 Dual 类型的分割基本单元，如果最大连接数量为 8，则对于每颗 FPGA，如 FPGA a 可以最多和另外 4 颗外部 FPGA 进行直接相连，且连接方向不限于水平和竖直两个方向；对于 Quad 类型的基本分割单元，如果其 max_int 为 16，则同理，每颗 FPGA 最大可与另外 4 颗外部 FPGA 进行直接相连，连接方向不限于水平和竖直。max_int 为最大对外连接的 FPGA 数量，实际连接 FPGA 数量可少于此值，实际连接通道将按实际值均分最大对外互连线数量。

对于每个FPGA，对外最大直接连接FPGA的数量为 $unit.max_int/2 = 4$



unit.ext_conn.num :600
 unit.int_conn.num :50
 unit.max_int: 8

对于每个FPGA，对外最大直接连接FPGA的数量为 $unit.max_int/4 = 4$



unit.ext_conn.num :1600
 unit.int_conn.num :600
 unit.max_int: 16

— 分割单元内部连线
 — 分割单元对外连线

连接方向

图 3 组网约束示意图

2、 #INT：指定在组网中，一颗 FPGA 最多与 #INT 颗其他 FPGA 通过连接线进行直接连接。

3、 #INT_DIST：指定在组网中，每颗 FPGA 与其他 FPGA 连线的距离

不得超过#INT_DIST 的大小。

动态组网部分的考核方式：给定 FPGA 使用数量#FPGA，以及代表用户设计的一组超图文件，要求程序能够在指定 FPGA 的数量的条件下完成组网并划分，求得最大的系统频率（至少保留小数点后两位有效数字）。根据实验经验及相关理论研究，最大系统频率在宏观上正比于公式右侧部分，公式右侧部分为真实系统频率的比例估算值：

$$sys_freq \propto \frac{100}{\frac{\#max_cut}{\#IO} + \#max_hop}$$

其中，#max-cut 为组网中所有 FPGA 对中最大的 cut size 大小，#IO 为对应产生最大 cut size 的两颗 FPGA 之间的互连线数量，#hop 为不具有直接物理连线连接支持的连线（net）路由途径的 FPGA 数量，#max-hop 为算法在对所有 net 进行 routing 后中间经过的 FPGA 数量的最大值，如图 4 示意，A,B,C 被划分至不同的 FPGA 中，连接它们的 net routing 路径如图中连线，其中 A,B 之间需要经过两颗 FPGA 进行中转，则其#hop=2；A,C 之间需要经过三颗 FPGA 进行中转，则其#hop=3，遍历所有 routing net 的#hop，其中的最大值为 #max-hop。

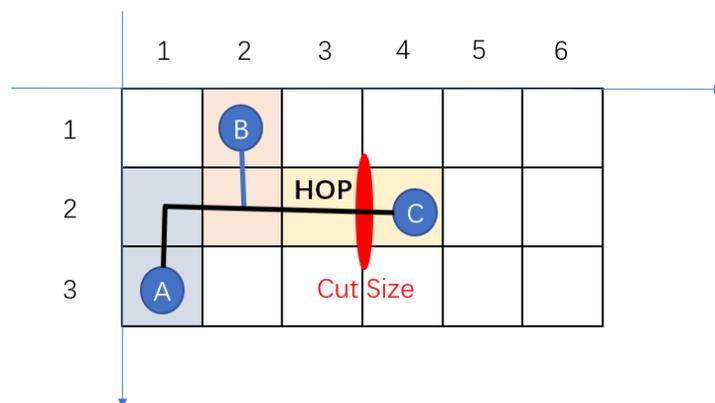


图 4 HOP 与 cut size 概念示意图

运行过程中的系统频率 `sys_freq` 与 `#max_cut` 需要在运行窗口中打印显示。

涉及知识点:

- 超图划分算法
- 路由算法
- 原型验证相关背景知识

六、评分标准

技术评分点:

- 组网方案与划分结果的完整性与正确性（主办方提供检查程序）
- 满足正确性的前提下，在同一平台（主办方将给出操作系统版本和编译器版本）上的运行时间，即所需的 CPU 算力
- 内存使用量
 - 使用 `top` 命令或工具 `memtime`
- 附加 Benchmark 测试（隐藏测试）的完成度

注：参赛队伍提供算法设计方案报告（含时间、空间复杂度分析）、可执行程序与源代码

a.技术评分，共 100 分。

1. 基础任务 – 60 分（Benchmark 测试的正确性占比 70%，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	划分基本单元为 single, 较少的节点(20)
testdata-2	划分基本单元为 single, 较少的节点(50)
testdata-3	划分基本单元为 single, 较少的节点(20) 和连线 (40)
testdata-4	划分基本单元为 single, 少量的节点(1000) 和连线 (1000)
testdata-5	划分基本单元为 dual, 大量的节点(12000) 和连线 (33000)
testdata-6	划分基本单元为 dual, 大量的节点(12000) 和连线 (14000)

2. 高性能任务 – 40 分。(Benchmark 测试的正确性占比 70%，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	划分基本单元为 quad, 大量的节点(12000) 和连线 (14000)
testdata-8	划分基本单元为 quad, 大量的节点(64000) 和连线 (64000)
testdata-9	划分基本单元为 quad, 大量的节点(1280000) 和连线 (1280000)
testdata-10	划分基本单元为 quad, 大量的节点(2560000) 和连线 (2560000)

说明：相关 FAQ 解答指南请联系 eda_ec@s2cinc.com。

七、参考资料 (赛题接口文件说明)

1.1 输入文件

- **Node definition file <节点定义文件 design.are>**

每个节点名称以字母 g 和一个不重复的数字组成，

每行表示一个节点的资源信息，包含 9 种资源，每种资源权重以 10 进制数值表示。

<node> PIO FF LUT BUFG TBUF DCM BRAM DSP PPC

例如，

常规节点文件，

g0 0 200 0 0 0 0 0 0 0

g1 0 200 0 0 0 0 0 0 0

g2 0 200 0 0 0 0 0 0 0

g3 0 200 0 0 0 0 0 0 0

g4 0 200 0 0 0 0 0 0 0

g5 0 200 0 0 0 0 0 0 0

- **Net definition file <连线定义文件 design.net>**

每个连线信息由 2 个或更多节点组成，一个为驱动节点(driver)，其他为负载节点(load)，

每行表示一个连线的部分信息。当两个节点之间有多条连线时，可以通过 weight 的大

小表示；也可以通过重复连线信息体现，此时权重即为累加值。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

例如，

g1 s 1

g0 l

g0 s 1

g2 l

g1 l

g0 s 1

g2 l

g1 l

- **The FPGA resource list file <FPGA 资源文件定义 design.info>**

单颗 FPGA 资源定义与节点资源定义一致。

格式如下，

<FPGA> PIO FF LUT BUFG TBUF DCM BRAM DSP PPC

如

FPGA: 1 109 1814 0 0 0 0 0 0

划分算法必须根据资源对图中节点进行划分，输出的结果不能满足资源约束时候输出错误报告，程序退出。

- **The FPGA Constraint file <FPGA 约束文件定义 design.fpga.constr>**

该文件列举了动态组网与划分中的约束条件，格式为：

FPGA.number :

unit.type :

unit.max_int:

unit.ext_conn.num :

unit.int_conn.num :

unit.ext_conn.max_dist :

FPGA Location: <fix node>

其中, FPGA.number 定义了动态组网中最多可以使用的 FPGA 数量; unit.type 为划分单元的类型, 可以为 single、dual、quad 中的一种; unit.max_int 规定一颗 FPGA 最多与 unit.max_int 颗其他 FPGA 通过连接线进行直接连接; unit.ext_conn.num 为该划分单元对外连接的互联线数量总数上限; unit.int_conn.num 为该划分单元内部互连线数量总数上限; unit.ext_conn.max_dist 为存在直接连接关系的 FPGA 之间的最大距离; FPGA Location 为划分基本单元中内部 FPGA 的位置编号, 固定节点 (fix node) 的划分位置被预先指定到某位置上, 如例子中 g20 和 g21 被指定到组网中任意一个 quad 型划分单元的 a 位置上, g22 被指定划分到任意一个 quad 型分割单元的 b 位置上。

例子如:

FPGA.number : 8

unit.type : quad

unit.max_int: 4

unit.ext_conn.num : 160

unit.int_conn.num : 120

unit.ext_conn.max_dist : 1

fpga.a : g20 g21

fpga.b : g22

划分算法必须根据约束进行划分，输出的结果不能满足资源约束时候输出错误报告，程序退出。

1.2 输出文件

- **The output partition result file <划分结果文件 design.output>**

划分结果文件 每一行包含一个分组信息里面包含的节点列表，以 'FPGA+坐标 分组+mm' 字符串开始，每个节点之间以空格隔开，每行建议最大包含 20 个节点，可以多行表示。TYPE 的值即为组网中 FPGA 的序数。FPGA 在组网中的坐标编号以左上方为原点，向右为横坐标正方向，向下为纵坐标正方向。

格式如下，

```
<FPGAm_n_p TYPE nn>: <node-list>
```

其中，m_n 代表 FPGA 的位置坐标，p 代表不同划分基本单元的 FPGA 位置，可选为 a,b,c,d.

例如，

```
FPGA1_1_a TYPE 1 :g0 g1 ..... g19
```

```
g20 g21 gp0
```

```
FPGA1_2_b TYPE 2 :g100 gp1 gp2
```

- **The output network design file <组网报告文件 design.network>**

组网部分为算法输出组网设计结构，格式为邻接矩阵，矩阵中 (ij) 位置的数值代表了

FPGA_i 与 FPGA_j 之间使用的物理互连线的数量。

举例如，

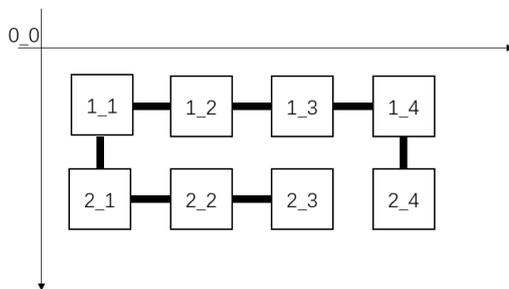
{ 0 100 120 130 }

{ 100 0 50 100 }

{ 120 50 0 10 }

{ 130 100 10 0 }

可以打印出组网可视图 (可选, 不作强制要求), 如



另外一部分为划分结果报告文件, 包含分组的实际资源权重使用值。

<FPGA_m TYPE nn>: <resource-list> [int-list]

{int-list} 采用 list 列表表示, 列表里面的值表示各分组之间的互连信号数量, 也包括经过的 routing net 的信号数量。

每个分组的内部互联权重值为 0, 不用考虑。

4 个分组的报告文件,

FPGA1_1_a TYPE 1: 1 400 109 1814 0 0 0 0 0 0 { 0 10 2 13 }

FPGA1_2_b TYPE 2: 2 350 0 4000 0 0 0 0 0 0 { 10 0 5 10 }

FPGA2_1_a TYPE 3: 1 200 109 1814 0 0 0 0 0 0 { 2 5 0 3 }

FPGA2_2_b TYPE 4: 2 350 0 4000 0 0 0 0 0 0 { 13 10 3 0 }

- **The routing result file <划分路由信息 design.route>**

该文件描述了算法经过 net routing 后的路由信息, routing 结果中最大 hop 距离和对应路径, 对应各 hop 距离的 routing net 数量, 每个 net 的编号, 起始节点名称, 路径等统计信息。

格式可以为

Max Hop: max hop number

Max Hop net id [start node end node][FPGA routing path]

Hop Distance1 net number

Hop Distance2 net number

...

Router Information:

net id [start node end node][FPGA routing path]

...

其中 net id 为 design.net 中的对应的 load 节点的文中行号, start node 和 end node 对应 design.net 中的节点名称, FPGA routing path 用 FPGA 编号组成路径; max hop number 为最大路由距离, 即路由经过的 FPGA 数量, 设定为除去终点起点节点所在的 FPGA 数量以后的中间 FPGA 数量。

例如,

Max Hop: 2

50 [g13 g34] [1,2] [2,3] [3,8]

1 23

2 12

...

Router Information:

8 [g17 g54] [1,2] [2,3]

...

其中 “50 [g13 g34] [1,2] [2,3] [3,8]” 代表 design.net 中第 50 行对应的 load 节点以及它对应的 driver 节点这条 net，起点为 g13，终点为 g34,路由路径为从 FPGA1->FPGA2->FPGA3->FPGA8。

“1 23

2 12” 代表 hop 距离为 1 的 routing net 数量为 23，hop 距离为 2 的 routing net 数量为 12，以此类推，计数各 hop 距离的 net 数量。