

一、 赛题名称

考虑时序驱动的设计划分器

二、 赛题背景

随着 FPGA 对更大逻辑容量和更高性能的需求不断提升，传统单片硅技术在良品率和工艺上遇到瓶颈。为突破这一限制，业内开始探索“化整为零”的方案，即将大规模逻辑划分为若干较小的单元，通过 2.5D 封装技术实现互联。这种堆叠硅片互联技术不仅提升了逻辑单元数量，也大幅度降低了功耗和散热问题，成为突破摩尔定律限制的重要手段。

如图 1 所示，在 2.5D 封装中各个 die 并不是垂直堆叠，而是侧向排列在一个共同的硅中介层上。这个中介层通常采用硅材料制成，具有与 die 相近的热膨胀系数，有助于降低热应力和提高封装的机械稳定性。图 2 为 die 间互联工艺示意图，图中各个 die 与硅中介层之间的连接采用微凸点技术。每个 die 的 I/O 信号通过微凸点焊接到中介层上，然后由中介层内的互联线将信号路由到其他 die 的对应端口。

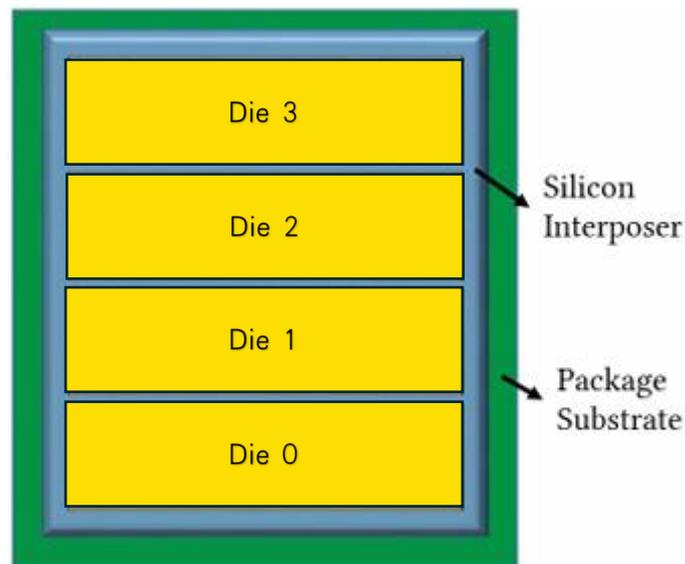


图 1. 2.5D FPGA 封装示意图

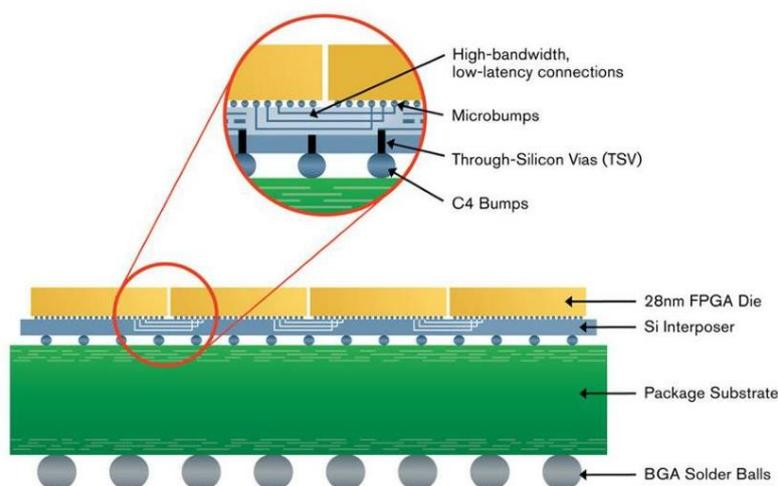


图 2. Die 间互联示意图

为适应 2.5D FPGA 的特点，FPGA 实现工具的一个核心环节是设计划分。其根据物理约束（包括 Die 内资源数量，die 间互联线数量，划分约束，IO 约束等）和时序约束，将设计中的子模块划分到各个 die 上。相比于传统的划分方法，考虑时序驱动的划分方案要求在满足各物理约束的前提下，得到时序优化的划分结果。

本赛题主要面向集成电路、计算机、数学等相关专业的学生，要求参赛者具有集成电路和 EDA 工具的相关知识和工程软件的开发能力，对于拥有时序驱动划分或 2.5D FPGA 系统设计实现经验的学生更加适合。

三、 赛题描述

本赛题要求参赛队伍根据给定的门电路网表、2.5D FPGA 架构信息、物理约束、时序约束，给出划分方案。以时序性能和程序运行时间作为划分质量的评价标准。

约束条件:

1. 每个测试用例将给出时序约束。如设计需满足目标频率（如 500MHz）。
2. 划分结果不应超过每个 Die 所包含逻辑资源（LUT/FF/DSP 等）的最高利用率
3. 每一个 die 间互联只能承载一根任意方向的 cut net, 划分结果不应超过 Die 间互联数的最高利用率
4. 划分算法不可改变输入文件与数据，不可修改网表结构，不可改变已固定的物理单元的位置

输入文件:

每个测试用例包含的输入文件如下:

1. 门级网表 (.edf)

该文件包含门电路网表, 该网表是按模块分级组织的。

2. 时序约束文件 (.sdc)

该文件包含所有相关的时序约束。

3. 物理约束文件 (.txt)

该文件将指定的 cell 和 port 约束到特定的 die 上, 格式如下:

```
cell_a die0
cell_b die1
cell_c die2
port_a die0
port_b die1
```

文件每一行为一条约束信息。每行包含 cell、port 名字及其被约束到的 die 名。该 cell 包含的所有子模块都应被约束到指定 die 上。

4. 2.5D FPGA 物理描述文件 (.xml)

该文件描述了目标 2.5D FPGA 各个 die 内的逻辑资源数、最高利用率, 以及 die 间互联数量和最高利用率。格式如下,资源类型将不限于 LUT, FF 等。

```
<?xml version="1.0" encoding="UTF-8"?>
<fpga>
  <!-- 定义各个 die 的逻辑资源-->
  <dies>
    <!--die 的名字-->
    <die name= "die0" >
      <!--资源类型、数量及最大利用率-->
      <resource type= "lut" count= "150000" util= "0.8" />
      <resource type= "ff" count= "150000" util= "0.8" />
      <resource type= "dsp" count= "150000" util= "0.8" />
      <resource type= "bram" count= "150000" util= "0.8" />
      <resource type= "carry" count= "150000" util= "0.8" />
    </die>
    <die name= "die1" >
      <resource type= "lut" count= "150000" util= "0.8" />
      <resource type= "ff" count= "150000" util= "0.8" />
      <resource type= "dsp" count= "150000" util= "0.8" />
      <resource type= "bram" count= "150000" util= "0.8" />
    </die>
  </dies>
</fpga>
```

```

</die>
<die name= "die2" >
  <resource type= "lut" count= "150000" util= "0.8" />
  <resource type= "ff" count= "150000" util= "0.8" />
  <resource type= "dsp" count= "150000" util= "0.8" />
  <resource type= "bram" count= "150000" util= "0.8" />
</die>
</dies>
<interconnections>
  <!-- die0 和 die1 之间的互联数量及最大利用率-->
  <interconnection dies= "die0 die1" count= "10000" util= "0.6" />
  <interconnection dies= "die1 die2" count= "10000" util= "0.6" />
</interconnections>
</fpga>

```

5. 时序文件(.lib)

该文件描述了各类 cell 的时序信息。下面的例子为 LUT6 的时序信息。

```

cell(LUT6) {
  area : 1;
  pin(I0,I1,I2,I3,I4,I5) {
    capacitance : INPUT_CAP;
    direction : input;
  }
  lut(L) {
    input_pins : "I0 I1 I2 I3 I4 I5";
  }
  pin(O) {
    direction : output;
    function : "L";
    timing() {
      timing_type : combinational;
      intrinsic_rise : 0.043;
      intrinsic_fall : 0.043;
      related_pin : "I0";
    }
    timing() {
      timing_type : combinational;
      intrinsic_rise : 0.043;
      intrinsic_fall : 0.043;
      related_pin : "I1";
    }
    timing() {

```

```

        timing_type : combinational;
            intrinsic_rise : 0.043;
            intrinsic_fall : 0.043;
            related_pin : "I2";
        }
    timing() {
        timing_type : combinational;
            intrinsic_rise : 0.043;
            intrinsic_fall : 0.043;
            related_pin : "I3";
        }
    timing() {
        timing_type : combinational;
            intrinsic_rise : 0.043;
            intrinsic_fall : 0.043;
            related_pin : "I4";
        }
    timing() {
        timing_type : combinational;
            intrinsic_rise : 0.043;
            intrinsic_fall : 0.043;
            related_pin : "I5";
        }
    }
}

```

线网 delay 计算说明:

所有线网的初始 delay 值统一估计为 0.2ns。线网每多一个 input 端口，delay 值增加 0.1ns。线网每次跨 die，则 delay 值增加 2ns。以图 3 为例，线网 a 的最终 delay 值应为 $0.2 + 3*0.1 + 2*2$ 。

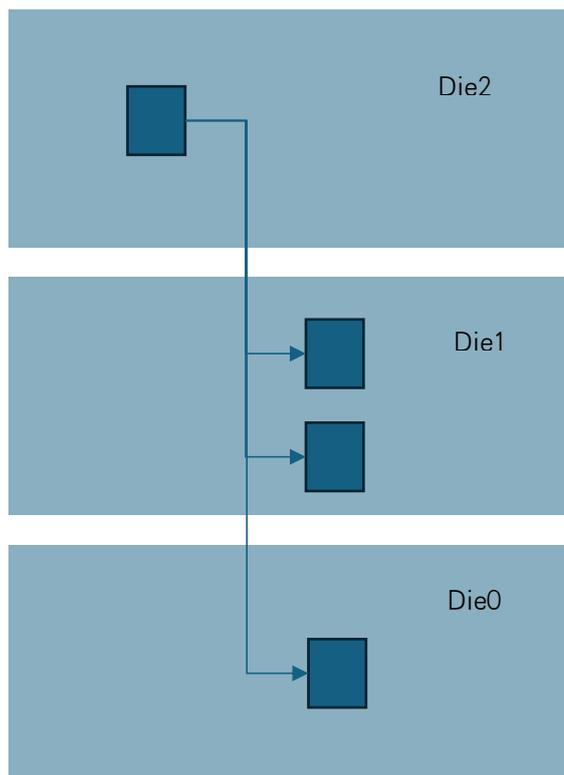


图 3 Net delay 计算

输出结果:

输出划分文件 (.txt) , 格式应如下:

```
cell_a die0  
cell_b die1  
cell_c die2  
port_a die0  
port_b die1
```

文件每一行为一条划分结果信息。每行包含 cell 名字及其所属 die 名。

四、 评分标准

本赛题将提供若干不同难度的测试用例。Hidden cases 的难度 与 Public cases 大致相当。验收时根据所有 cases 的总分进行排名。大赛将提供评估程序, 供参赛队伍计算 TNS, WNS。

1. 评分标准:

- 1) 若输出结果违反任意约束，该用例计 $\alpha + \beta + \gamma$ 分。
- 2) 按如下函数对每个 test case 进行评分，并依所有 test cases 总分进行排名。

$$Score = \alpha * \left| \frac{WNS}{WNS_{MAX}} \right| + \beta * \left| \frac{TNS}{TNS_{MAX}} \right| + \gamma * \frac{Runtime}{Runtime_{MAX}}$$

α 、 β 、 γ 为权重，将在后续说明中给出具体值。

WNS (最差负时序松弛)：反映设计中关键路径的时序违规程度，绝对值越大表示时序问题越严重。

WNS_max: 为所有竞赛结果中最差的 WNS (如-200ps)

TNS (总负时序松弛)：反映所有时序路径的累积违规程度，绝对值越大表示整体时序收敛性越差。

TNS_max: 为所有竞赛结果中最差的 TNS (如-5000ps)

Runtime (运行时间)：优化过程耗时，单位为秒。

Runtime_max: 为所有竞赛结果中最长的运行时间 (如 24 小时=86400 秒)

2. 提交说明:

要求提供可执行文件，提供讲解 PPT，鼓励提供源代码。所依赖的第三方库也需一并提供。

五、 参考资料

[1] Chirag Ravishankar;Dinesh Gaitonde;Trevor Bauer, "Placement Strategies for 2.5D FPGA Fabric Architectures", 2018 28th International Conference on Field Programmable Logic and Applications (FPL)

[2] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, 'Multilevel Hypergraph Partitioning: Application in VLSI domain', Proc. ACM/IEEE DAC, June 1997.

[3] OpenSTA: <https://github.com/parallaxsw/OpenSTA>

*本赛题指南未尽问题请参加竞赛宣讲会或见赛题 Q&A 文件。