2024中国研究生创“芯”大赛·EDA精英挑战赛

赛题指南

一、 赛题名称 Title

考虑时序驱动布局的芯粒划分器

Chiplet Partitioner with Timing Driven Placement

二、 命题单位 Company

上海合见工业软件集团有限公司

Shanghai UniVista Industrial Software Group Co., Ltd.

三、 赛题主席 Contest Chair

朱可人（香港中文大学）

Keren Zhu, Chinese University of Hong Kong

四、 赛题背景 Problem Background

随着半导体进入后摩尔时代，晶圆工艺的发展遇到了瓶颈，导致芯片性能的提升无法再依赖先进工艺的自然演进来推动。集成电路制造面临着不断增加的复杂性和单一制程工艺的限制，导致设计周期延长、成本上升和制造风险增加。

芯粒技术是一种新的芯片设计和封装技术，它通过将片上系统（SoC）分成较小的芯粒，分别选择适合的制程工艺进行制造，再通过新型封装技术，将不同功能、不同工艺制造的芯粒彼此互联，最终集成封装为一个系统级芯片组，从而提高良率和降低成本，同时提高设计的灵活度，降低设计周期。在芯粒技术下，如何将设计电路划分给不同的芯粒成为了新的技术突破点。芯粒划分结果会影响整个芯片的性能，功耗，面积以及成本。

芯粒设计的一个核心环节是顶层芯片的芯粒划分，其根据芯粒约束（包括芯粒数量，划分约束，面积约束等）和时序约束，进行芯粒的划分。相比于传统的划分方法，考虑时序驱动布局的芯粒划分方案要求在满足各芯粒约束的前提下，得到时序优化的全芯片布局，并进行芯粒划分，在规模和约束条件上对布局与划分提出了新的挑战。

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，对于拥有时序驱动布局或多芯粒系统设计实现经验的学生更加适合。

With the advent of the post-Moore's Law era in semiconductors, the development of wafer processes has encountered bottlenecks, impeding further performance gains from the natural evolution of advanced processes. Integrated circuit manufacturing faces escalating complexity and constraints from single-process technology, leading to extended design cycles, rising costs, and increased manufacturing risks.

Chiplet technology is a novel approach to chip design and packaging. It involves breaking down System-on-Chip (SoC) into smaller chiplets, each manufactured using suitable process technologies. These chiplets, featuring different functionalities and fabricated using diverse processes, are interconnected using advanced packaging techniques and integrated into a system-level chip group. This enhances yield rates, reduces costs, boosts design flexibility, and decreases design cycles. In this technology, the partitioning of design circuits into different chiplets has emerged as a pivotal technological breakthrough. The result of the chiplet partitioning profoundly impacts the overall chip's performance, power consumption, area utilization, and cost-effectiveness.

One of the core aspects of chiplet design is the top-level partitioning scheme of chiplets, which partitions chiplets across the entire chip based on chiplet constraints (including number of chiplets, partitioning constraints, area constraints, etc.) and timing constraints. Compared to traditional partitioning methods, the chiplet partitioning scheme with timing driven placement requires globally optimizing the chip placement with timing considerations, while adhering to individual chiplet constraints, and then performing the chiplet partitioning. This approach presents new challenges in terms of scale and constraints for both placement and partitioning.

This topic is suitable for students majoring in Integrated Circuits, Computer Science, Mathematics, etc. Participants are required to have relevant knowledge of integrated circuits and EDA tools, along with proficiency in engineering software development. Students with experience in timing-driven placement or multi-chiplet system design and implementation are particularly suitable for this topic.

五、 赛题描述 Description of Problem

本赛题要求参赛队伍根据给定的全芯片门电路网表、布图约束、时序约束、物理约束以及芯粒划分约束，给出全芯片的布局结果和在此基础上的芯粒划分方案。以时序性能作为布局划分质量的评价标准。

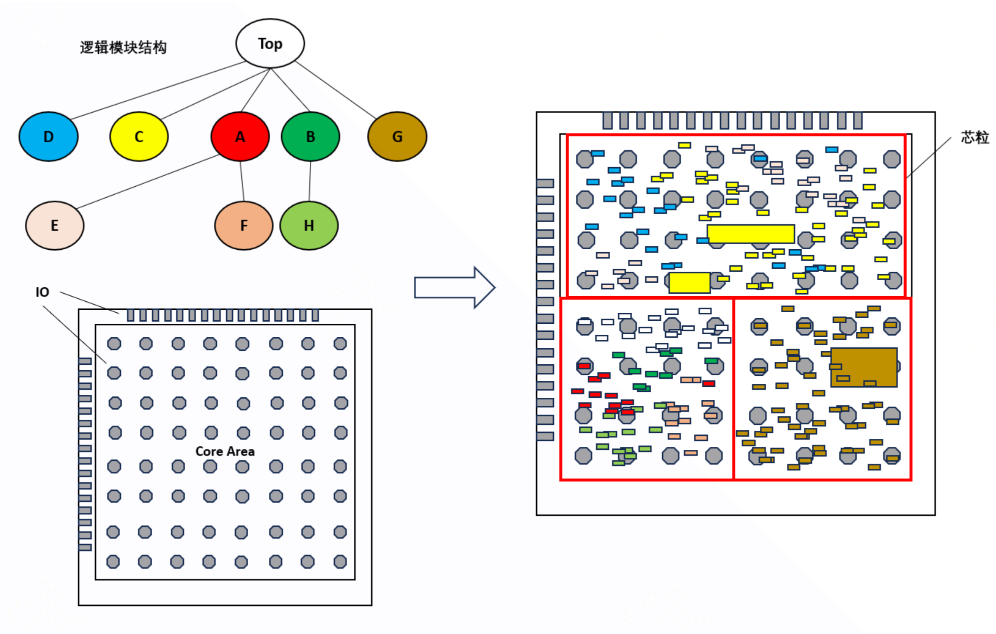

图 1. 赛题示例

This topic requires participating teams to provide a placement solution for the entire chip based on a given gate-level netlist, floorplan constraints, timing constraints, physical constraints and chiplet partitioning constraints. Additionally, teams must propose a chiplet partitioning scheme based on this placement. The evaluation criteria will primarily focus on the timing performance of the placement partitioning quality.

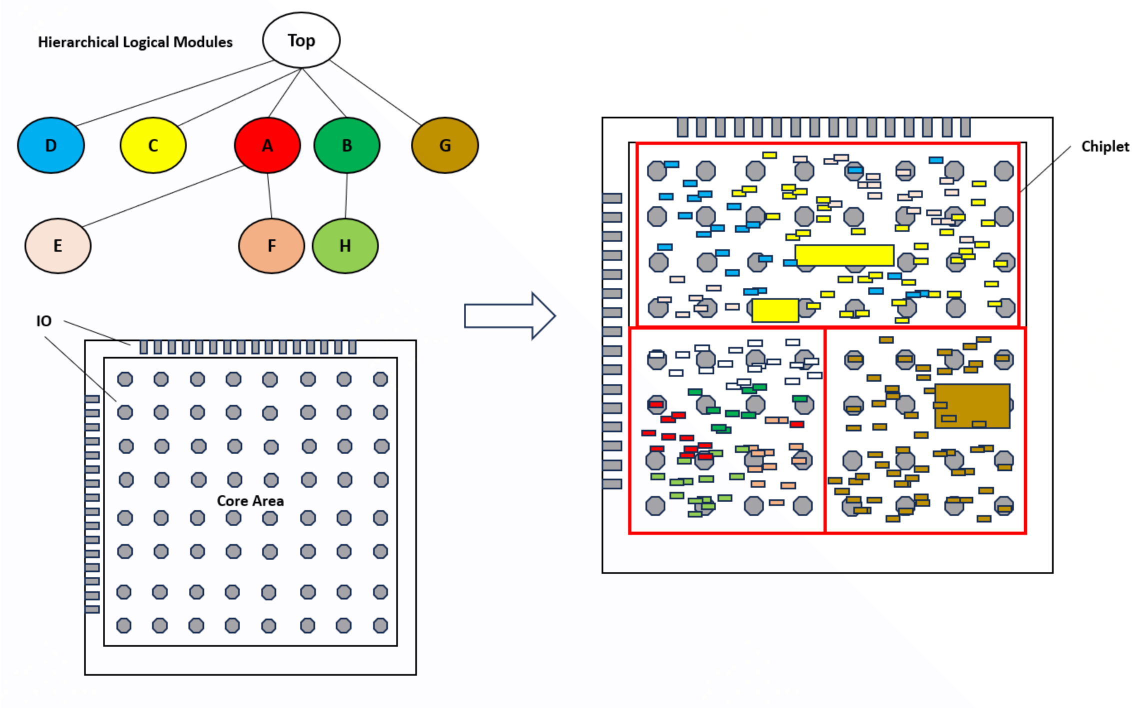

Figure 1. An example of task

约束条件：

1. 对全芯片中的所有物理单元在 core area 内进行布局，且重叠率<5%，重叠率 $= \frac{\text{重叠面积}}{\text{物理单元总面积}}$，多次重叠部分的面积将被多次统计。
2. 不同芯粒的划分区域应互不重叠，且不超出 core area。
3. 每个芯粒的形状都为矩形，并满足最小面积约束和一定范围的利用率和长宽比。
4. 按逻辑模块划分芯粒，所有模块都应可以确定其唯一所属芯粒，请参看划分约束文件。
5. 划分约束中要求属于同一芯粒的模块应划分为同一芯粒。
6. 布局结果中，每个模块的标准单元和宏单元，应完全位于该模块所属的芯粒区域内。
7. 不可改变输入文件与数据，不可改变已固定的物理单元和芯片 IO 的位置。
8. 运行时间与峰值内存不可超过对应测试用例的要求。

Constraints:

1. Place all physical cells of the entire chip within the core area, with an overlap rate <5%. The overlap rate is defined as the overlap area divided by the total physical unit area, and areas overlapping multiple times will be counted multiple times.
2. Partitioning regions of different chiplets should not overlap and must stay within the core area.

3. The shape of each chiplet is rectangular and must satisfy the minimum area constraint and certain utilization and aspect ratio ranges.
4. Partition the chiplets based on logical modules. Each module should be partitioned into one chiplet, but its submodules can be partitioned into another chiplet. Please refer to the partitioning constraints file.
5. The modules belonging to the same chiplet from the partitioning constraints must be partitioned into the same chiplet.
6. The valid placement result should ensure all the standard cells and macro cells are placed within the chiplet's region.
7. The input file and data must not be changed. Locations of fixed cells and I/O pins must not change.
8. The runtime and peak memory must not exceed the requirements of the corresponding test cases.

输入文件：

1. 全芯片门电路网表 Verilog 文件(.v)

   该文件包含全芯片的门电路网表，该网表是按模块分级组织的。本赛题不会使用包含多实例化模块（MIM）的网表。

2. 工艺文件(.lef .lib)

   该文件中包含设计中所涉及的技术参数和时序模型。

3. 布图文件(.def)

   该文件包含布局前的floorplan数据和给定的芯片IO的位置。

4. 时序约束文件(.sdc)

该文件包含所有相关的时序约束

5. 物理约束文件(.txt)

该文件包含 core area 的尺寸、任意芯粒的最小面积、每个芯粒的利用率和长宽比约束，格式如下：

```
0 0 1000 1000
100000
0.5 0.8 1.0 1.0
0.5 0.7 0.67 1.5
0.6 0.8 0.5 2.0
```

该文件共 N+2 行，N 为目标芯粒的数量；第一行的 4 个整数用空格分隔，每两个一组，依次表示 core area 的左下和右上坐标；第二行表示任意芯粒的最小面积；从第三行开始，每行的 4 个 float 数值依次表示芯粒的最小利用率、最大利用率、最小长宽比，最大长宽比，用空格分隔。

6. 划分约束文件(.txt)

该文件包含芯粒的划分约束，格式如下：

```
ModuleA ModuleB ModuleTop
ModuleD ModuleA/E
```

每行表示一组模块的划分约束，同一行用空格分隔的模块，要

求必须划分在任意同一芯粒。每行中的模块名称代表该模块及其所有子模块，若其任意子模块出现在其它行，则此名称视为除去该子模块的部分。ModuleTop 仅代表顶层模块，不包含任何其子模块。注意，不同行的模块组，可以划分在同一芯粒，也可以划分在不同芯粒。文件 6 中各行模块组同文件 5 中第 3 行到第 N+2 行物理约束无任何对应关系。

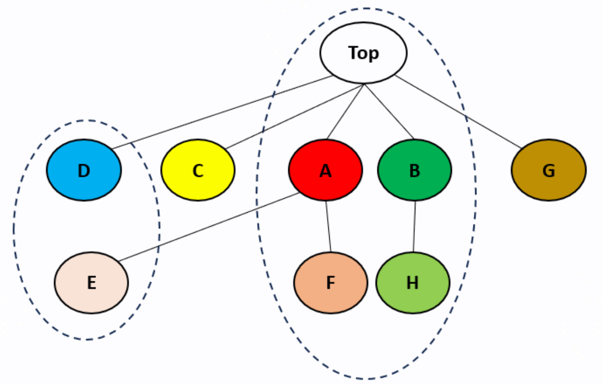

图 2. 划分约束示例

Input Files:

1. Full-chip gate-level netlist Verilog file (.v)

   This file contains the gate-level netlist of the entire chip, organized hierarchically by modules. The netlist does not include any Multiple Instantiation Module (MIM).

2. Technology files (.lef .lib)

These files include the technology data and timing models relevant to the design.

3. Floorplan file (.def)

   This file contains floorplan data prior to placement and specifies the locations of the chip's I/O pins.

4. Timing constraints file (.sdc)

   This file includes all the timing constraints.

5. Physical constraints file (.txt)

   This file contains the coordinates of the core area, constraints on minimum area of any chiplet, utilization and aspect ratio for each chiplet, formatted as follows:

```
0 0 1000 1000
100000
0.5 0.8 1.0 1.0
0.5 0.7 0.67 1.5
0.6 0.8 0.5 2.0
```

   The file consists of N+2 lines, where N is the number of target chiplets. The first line contains 4 integers separated by spaces, grouped into pairs, representing the coordinates of the core area's bottom-left and top-right corners. The second line indicates the minimum area for any chiplet. Starting from the third line, each line contains 4 float values separated by spaces, representing respectively the minimum utilization, maximum utilization, minimum aspect ratio, and maximum aspect ratio of each chiplet.

6. Partitioning constraints file (.txt)

This file contains the partitioning constraints, formatted as follows:

```
ModuleA ModuleB ModuleTop
ModuleD ModuleA/E
```

Each line represents a set of partitioning constraints for modules. Modules listed on the same line are required to be partitioned onto one single chiplet. Except for ModuleTop, each module name represents that module and all its submodules. If any of its submodules appears in another line, then this name is considered as excluding that submodule. ModuleTop only represents the top-level module and does not include any of its submodules. Please note that modules from different lines can be partitioned on a same chiplet or on different chiplets. There is no correspondence between the module groups in file 6 and the physical constraints of the 3rd ~ N+2 lines in file 5.

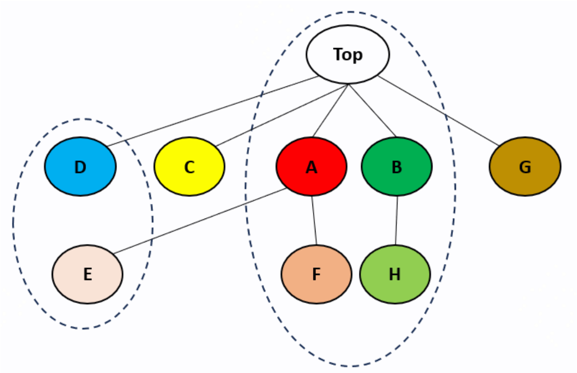

Figure 2. An example of partitioning constraints

输出文件：

7. 布局文件(.def)

该文件用于指定布局完成后的物理单元摆放数据。

8. 芯粒划分文件(.txt)

该文件用于指定每个芯粒的形状与位置信息，以及每个芯粒所包含的模块，格式如下：

```
0 0 500 500
500 0 1000 500
10 500 990 1000
ModuleTop ModuleB ModuleA
ModuleG
ModuleD ModuleA/E ModuleC
```

文件中前N行依次表示物理约束中对应行芯粒的区域（即文件8第1行芯粒区域对应于文件5第3行芯粒物理约束；文件8第2行芯粒区域对应于文件5第4行芯粒物理约束，以此类推），4个整数用空格分隔，每两个一组，依次表示该区域的左下和右上坐标；第N+1到2N行，依次表示对应芯粒所包含的模块（即第N+1行模块组对应于第1行芯粒区域，第N+2行模块组对应于第2行芯粒区域，以此类推），模块名称含义与划分约束文件（文件6）相同。

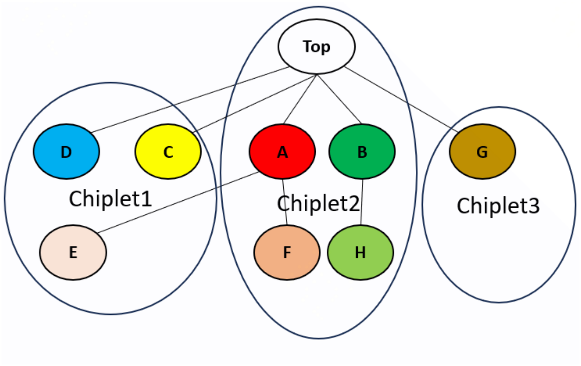

图 3. 划分方案示例

Output Files:

7. Placement file (.def)

   This file is required to specify the placement data of physical cells after placement.

8. Chiplet partitioning file (.txt)

   This file is required to specify the shape and position information of each chiplet, as well as the modules contained in each chiplet, formatted as follows:

```
0 0 500 500
500 0 1000 500
10 500 990 1000
ModuleTop ModuleB ModuleA
ModuleG
ModuleD ModuleA/E ModuleC
```

The first N lines in the file sequentially represent the regions of each chiplet, corresponding to lines in the physical constraints file (i.e., the chiplet region on line 1 of file 8 corresponds to the chiplet physical constraints on line 3 of file 5; the chiplet region on line 2 of file 8 corresponds to the chiplet physical constraints on line 4 of file 5, and so on). Each line consists of 4 integers separated by spaces per line. Each pair of integers indicates the bottom-left and top-right coordinates of the region. Lines N+1 to 2N sequentially represent modules contained within the corresponding chiplet (i.e., line N+1 corresponds to the region on line 1 of the chiplet, line N+2 corresponds to the region on line 2 of the chiplet, and so forth). The module name has the same meaning as in the partitioning constraints file (File 6).

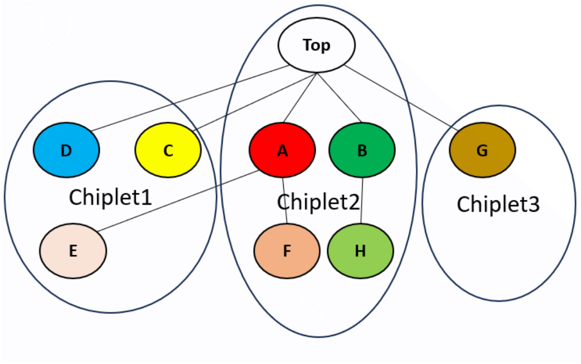

Figure 3. An example of partitioning scheme

六、 评分标准 Scoring criteria

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。Public cases 仅供练习与过程测试使用，验收时根据所有 Hidden cases 的总分进行排名。比赛开始后公布每个用例的分值、最长运行时间和最大峰值内存，并提供赛题评估程序，供参赛队伍计算 TNS，WNS，运行时间和峰值内存。

This contest will provide several test cases of varying difficulties. The difficulty of hidden cases is roughly equivalent to that of public cases. Public cases are provided for practice and process testing purposes only. Ranking during evaluation will be based on the total score of all hidden cases. After the contest begins, the case score, maximum runtime, and peak memory of each test case will be disclosed, along with providing an evaluation program for teams to compute TNS (Total Negative Slack), WNS (Worst-case Negative Slack), runtime and peak memory.

1. 评分标准 Scoring criteria

1) 若输出结果违反任意约束，该用例计 0 分。

2）按如下公式对每个 Hidden case 进行评分，并依所有 Hidden cases 总分进行排名。

$$s = C\cdot\frac{max\left(36+10\frac{\mathrm{TNS}_r^l-\mathrm{TNS}^l}{\mathrm{TNS}_r^l}+2\frac{\mathrm{TNS}_r^e-\mathrm{TNS}^e}{\mathrm{TNS}_r^e}+5\frac{\mathrm{WNS}_r^l-\mathrm{WNS}^l}{\mathrm{WNS}_r^l}+\frac{\mathrm{WNS}_r^e-\mathrm{WNS}^e}{\mathrm{WNS}_r^e},\ 0\right)}{54}\cdot\frac{(16+\min\left(\max\left(log_2\frac{t_r}{t},-4\right),4\right))}{20}$$

上述公式中s表示本用例的得分，C表示本用例的分值（不同用例的分值可以不同），t表示运行时间；上标l表示 late 指标，上标e表示 early 指标；下标r指本用例该指标的参考值。

1) If the output violates any constraints, the test case scores zero points.
2) Each hidden case is scored according to the following formula, and teams are ranked based on the total score of hidden cases.

$$s = C\cdot\frac{max\left(36+10\frac{\mathrm{TNS}_r^l-\mathrm{TNS}^l}{\mathrm{TNS}_r^l}+2\frac{\mathrm{TNS}_r^e-\mathrm{TNS}^e}{\mathrm{TNS}_r^e}+5\frac{\mathrm{WNS}_r^l-\mathrm{WNS}^l}{\mathrm{WNS}_r^l}+\frac{\mathrm{WNS}_r^e-\mathrm{WNS}^e}{\mathrm{WNS}_r^e},\ 0\right)}{54}\cdot\frac{(16+\min\left(\max\left(log_2\frac{t_r}{t},-4\right),4\right))}{20}$$

In the above formula, s represents the score obtained, C represents the maximum score for this test case, and t represents the execution time. The superscript 'l' denotes the 'late' metric, and the superscript 'e' denotes the 'early' metric. The subscript 'r' indicates the reference value of this metric.

2、提交说明 Submission Instructions

要求提供可执行文件，所依赖的第三方库也一并提供。

Please provide the executable file along with the third-party libraries it depends on.

七、 参考资料：References

[1] C.-K. Cheng, A. B. Kahng, I. Kang and L. Wang, "RePlAce: Advancing Solution Quality and Routability Validation in Global Placement", (.pdf), IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 38(9) (2019), pp. 1717-1730.

[2] Y. Lin et al., "DREAMPlace: Deep Learning Toolkit-Enabled GPU Acceleration for Modern VLSI Placement," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 40, no. 4, pp. 748-761, April 2021, doi: 10.1109/TCAD.2020.3003843.

[3] Peiyu Liao, Dawei Guo, Zizheng Guo, Siting Liu, Yibo Lin, Bei Yu, "DREAMPlace 4.0: Timing-Driven Placement With Momentum-Based Net Weighting and Lagrangian-Based Refinement", IEEE Transactions on Computer-

Aided Design of Integrated Circuits and Systems, vol.42, no.10, pp.3374-3387, 2023.

[4] Tsung-Wei Huang and Martin Wong, "OpenTimer: A High-Performance Timing Analysis Tool," IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 895-902, Austin, TX, 2015

*本赛题指南未尽问题，见赛题 Q&A 文件

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