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：AI-Based 的智能版图拼接

二、 命题企业：北京华大九天科技股份有限公司

三、 赛题 Chair： 林亦波 北京大学、贺旭 湖南大学

四、 赛题背景：

模拟 IC 电路设计一般属于 EDA 自动化流程的后端，为提升模拟集成电路设计的自动化和智能化程度，需建立模拟集成电路从性能指标到电路模型自动建立、版图自动生成，提高模拟集成电路自动化设计和分析的效率。而模拟 IC 电路的设计难以摆脱人工，因有以下挑战：

1. 结构复杂、参数、设计规则以及指标多，搜索空间庞大
2. 器件级的拓扑结构与布线复杂多变
3. 极度依赖人的经验积累、研发周期长

对于模拟电路的设计规则，需要考虑的因素很多，譬如 PPA（power ，performance，area），电路对称，匹配，分组，电路寄生，连接寄生，温度，电源，频率甚至工艺参数等；如果所有的约束都考虑，整体的设计框架将十分复杂，鉴于大赛时间周期的考虑，本次赛题仅考虑模拟电路的一个局部的场景，设计规则专注于线长，面积和布线成功率。在设计的过程中，常常遇到如下版图拼接的情形：

在 IC 设计的角落位置已经摆放了其他器件，在剩余的空间区域内，需要摆放一些电路元件，即对于给定数目的多个不同形状和尺寸的芯片版图，

根据给定规则，对其进行合理摆放，使模块可以在限定区域摆放成功，并且满足模块布线的要求。

在这样的设计场景下，人工版图拼接的工作量也是负担沉重，由于仿真各种指标的时间较长，导致设计迭代周期长，而且手动拼接利用率也难达最优。为了提升模拟电路设计的时间，加速设计的开发周期，因此需要 AI 学习加入以提升优化速度，让模拟自动布局的时间达到实时效果。同时解决该问题的 AI 方法也可方便扩展到其他有关 AI Placement 的应用，前景十分广泛。

五、 赛题描述：

赛题需对给定连接关系和布局区域的模块进行 AI 加数学优化驱动的布局，出题方的布线评价程序计算得到每个布局的布通率、面积及线长等指标，要求时间为秒级完成一次合格的布局。本赛题考察如何使用神经网络、机器学习或者强化学习等 AI 算法，结合优化算法，对布局进行优化，提升模块布局的速度，提高模块布局的布通率指标，高效得到指标优秀的模块布局。时间、面积，线长和布通率将是主要考察指标。

1）输入：给定 N 个不同形状和尺寸的多边形模块，多边形为矩形或边均为正交方向的多边形， 给定摆放的多边形限制区域以及模块之间的端口连接关系。(判定多边形在限定区域是否可以布线成功由华大提供接口支持。为了方便进行各种预测网络的训练，华大将提供至少二万个样本，包括模块位置、端口信息、连接关系、布通率、成功布线的线长，面积以及布线失败的模块等信息。 每个样本对应的布局内模块数量从 5 个到 50 个不等，方便学生循序渐进不断打破技术极限。)

2）求：限定区域内的模块拼接的最小面积结果

3）拼接规则：

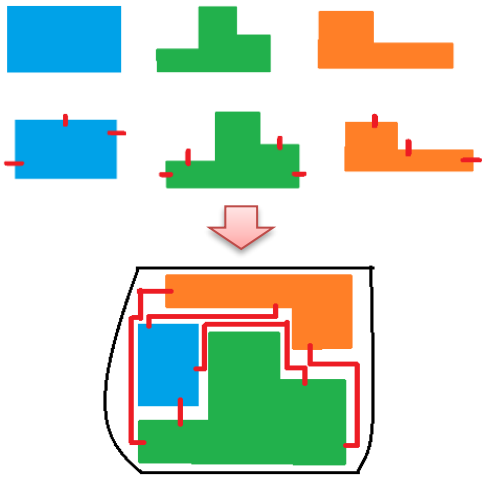
a) 各多边形之间不允许重叠（overlap）

b) 各多边形允许做基本的几何旋转(0°/90°/180°/270°/MX/MY/MXR90/MYR90)

c) 摆放多边形不可以超出限定的区域，否则视为无效多边形

d) 多边形拼接的有效性，需通过布线接口验证是否可以布线成功

4 ）举例：对以下 3 个多边形，如下的拼接方式， 在限定的区域内摆放所有多边形，布线成功，总线长相对较短，模块和布线围成的面积相对较小。



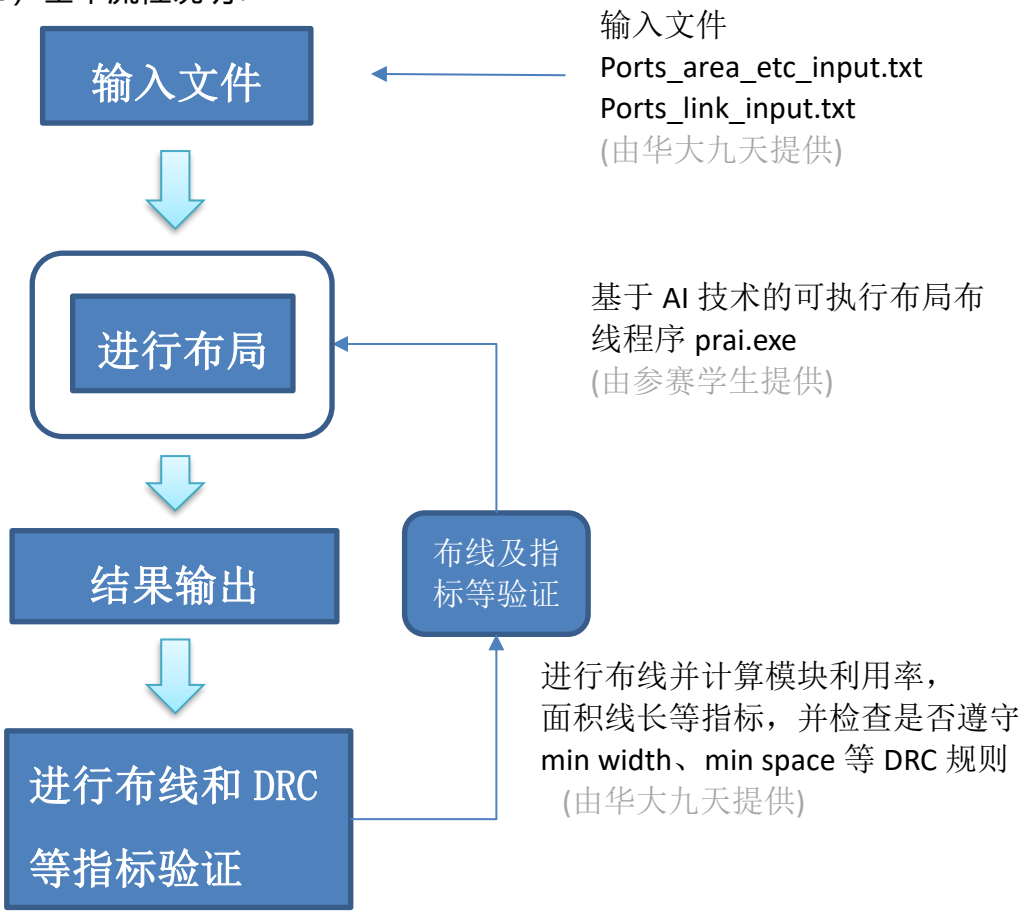
5）布线器布线规则说明：

a) 有连接关系的模块尽量放在一起

b) 端口距离较近的优先连接

c）其他的布线在剩余空间搜索布线路径，如果布线失败，引擎将受影响的布线拆掉，采取拆线的策略进行重新布线，如果尝试一定次数，如果布线还是不成功，宣告失败，结果以布线成功条数多的结果为准。

6）基本流程说明：



最优布局的算法总体求解流程如上，打分流程将对输出结果进行验证，并输出布局的结果指标，评分按照打分规则进行。

由于布局时间的限制为秒级，学生需提供基于 AI 和数学优化驱动的布局程序，快速得到高质量的布局结果。

验证程序输出指标主要包括：布线成功模块的个数，布线成功模块的面积占比，模块和布线围成的面积及线长

7）格式文件说明：

模块文件格式 Ports_area_etc_input_1.txt：

```
Area: (XXX, XXX) (XXX, XXX) (XXX, XXX) (XXX, XXX)     // 布线区域设置
Rule: SD (5, 5); GATE (4, 4); SD_GATE (0.5); SD_ITO (0.5); GATE_ITO (0.5)
Module：M1                                            //模块 M1
Boundary：(0, 0) (90, 0) (90, 90) (0, 90); GATE       //M1 的边界
Port:   (0, 45) (5, 45) (5, 50) (0, 50); SD           //M1 的第一个端口，颜色 SD
Port:   (45, 0) (45, 5) (50, 5) (50, 0); GATE         //M1 的第二个端口，颜色 GATE
Module：M2
Boundary：(XXX, XXX)  (XXX, XXX)  (XXX, XXX)  (XXX, XXX); ITO
Port:   (XXX, XXX) (XXX, XXX) (XXX, XXX) (XXX, XXX); SD
Port:   (XXX, XXX) (XXX, XXX) (XXX, XXX) (XXX, XXX); GATE
```

连接文件格式 Ports_link_input_1.txt：

```
Link1:
M1    M2    M3
1      1      3    //模块 M1 的第一个端口，与 M2 的第一个端口和 M3 的第三个端口连接在一起
Link2:
M5    M7
2      3          //模块 M5 的第二个端口，与 M7 的第三个端口连接在一起
Link3:
M3    M5
1      2
```

摆放输出模块位置文件格式 result_1.txt：

```
Module：M1                                          //模块 M1
Orient：R0/R90/R180/R270/MX/MY/MXR90/MYR90          //M1 摆放位置的旋转方向
Position :( XXX, XXX)                               //M1 摆放的中心点位置
Module：M2
Orient：R0/R90/R180/R270/MX/MY/MXR90/MYR90
Position :( XXX, XXX)
...
```

8）提交要求：

1. 源工程项目代码
2. 服务器上可编译执行程序prai.exe
3. 一份设计报告，说明算法设计和测试效果。

其中，对可执行程序 prai.exe 的要求是：

a）Linux 服务器环境上可以正确执行

b）执行示例：prai.exe /xxx/xxx/Ports_area_etc_input_<id>.txt /xxx/xxx/Ports_link_input_<id>.txt

c）输出：/xxx/xxx/result_<id>.txt

d）说明：

- 输出文本在用户特定目录下
- 输出文本命名有要求，必须是 result_<id>.txt
- id 指的是 Case 的序号

9）求解思路猜想：

本次赛题强调的是基于 AI 的智能版图拼接，目前比较热门的研究领域的技术如下：

1. 利用强化学习并结合 GCN 技术完成端到端的自动化布局设计，并使布局功能具有较强的迁移性。（迁移性指的是支持不同数量的模块连接关系）

2. 启发式算法/数学优化算法与机器学习结合，让机器学习更好地对启发式算法/数学优化算法的局部进行加速和优化。

3. 多智能体结合的机器学习技术，让布局效率和质量更佳。

4. 利用 routing-aware driven 技术指导布局，加速布局的效率和质量。

5. 结合 Transformer 网络和 GCN 技术，快速预测布局质量，提升布局的速度和结果。

6. 参考文献，见本文最后章节

六、 评分标准

a）打分测试 Case 说明：Case 模块数量从 5 个到 70 个， 模块数量在

25-50 个之间占主要比重。打分 Case 的数量初步定在 40-50 个，将根据实际比赛的情况酌情增减。

b）布局结果：每个 case 分数包含三部分

（1）全部放入限定区域并且布线成功分数

不能全部放入，分数等于 Case 布线成功分数*模块利用率*模块面积利用率

模块利用率 = 成功模块数/总模块数量

模块面积利用率 = 成功模块面积/总模块面积

说明： 如果在限定区域放入较多模块，但是通过华大提供的布线接口检查，发现有些多边形接口连接失败，失败的模块将不计入成功个数和面积；如果摆放模块存在重叠，重叠模块都将不计入利用率结果。

布线成功分数根据模块的数量而定：

25 个模块以下：50 分；

25-50 个模块：70 分；

50 个模块以上：90 分；

由于机器学习算法很难每个 Case 布局后都布线成功，根据不同 Case 数量设定不同的标准，有机会得到后面的面积和线长分数。

25 个模块以下：最多两个模块失败

25-50 个模块：90%模块成功率

50 个模块以上：85%模块成功率

（2）全部成功放入的布线模块和布线围成的面积按照排序打分，面积越小分数越高，20 分/17 分/15 分/13 分/11 分/9 分/7 分/5 分/3 分/1

分， 其他零分；如果满足设定的标准，有不成功的模块，打分的面积 = 成功模块和布线围成的面积/模块面积利用率

（3）全部成功放入的模块布线的总线长按照排序打分，总线长越短分数越高10分/9分/8分/7分/6分/5分/4分/3分/2分/1分,其他零分；如果满足设定的标准，有不成功的布线模块，打分线长 = 成功模块的总布线长度/布线连接成功率

b）运行时间：20 分

（1）模块数量 50 个以下 15 秒内不出结果，技术得分判 0

（2）模块数量 50 个以上 25 秒内不出结果，技术得分判 0

（3）时间技术分将按照比赛的成绩，划分为四档：A（20 分）， B（15 分）， C（10 分）， D（5 分）

(4) 线程不超过 8 个（调用布线评价程序线程）

(5) 模型学习训练时间不计入考核

c）九天将根据评分规则的不足，逐步改进规则，以利于提升比赛水平

七、 参考资料

1) TR Lin，D Penney，M Pedram，L Chen: A Deep Reinforcement Learning Framework for Architectural Exploration: A Routerless NoC Case Study. 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)

2) A Mirhoseini，A Goldie，M Yazgan，JW Jiang，J Dean: A graph placement methodology for fast chip design. 10.1038/s41586-021-03544-w

3) A Goldie，A Mirhoseini: Placement Optimization with Deep Reinforcement Learning. 2020

4) Azalia Mirhoseini, Anna Goldie, Google：Chip Placement with Deep Reinforcement Learning. 22 Apr 2020. https://arxiv.org/abs/2004.10746v1

5) Otten, R.; Ginneken, L. P. P. P.: Floorplan Design Using Simulated Annealing. Prov. Of the ICCAD 1984, 96-98, 1984

6) Ritter, H.; Schulten, K.: Topology Conserving Mappings for Learning Motor Tasks. in Neural Networks for Computing, AIP Conference Proceedings 151, Ed. J.S.Denker, pp.376-380, Snowbird, Utah, 1986

7) Zhang, C.; Mlynski, D. A.: VLSI-placement with a neural network model. Proc. Int. Symp. on Circ. and Syst. , pp. 475-478, 1990.

8) Aykanat, C.; Bultan, T.; Haritaoglu, I.: A fast neural-network algorithm for VLSI cell placement’ , NeuralNetworks, Vol. 11, No. 9, pp.1671–1684. 1998

9) Hopfield, J.; Tank, D.W.: Neural computation of decisions in optimization problems

10) H .; F u ji y o s h i, K .; K a jit a n i, Y .: VLSI module placement based on rectangle packing by the sequence-pair,’ ‘ IEEE Trans. on Computer-Aided Design of Integrated Circuits and Sys-tems,

Vo1.15, No.12, pp.1518-1524 (1996).

11) MCNC Benchmark Netlists for Floorplanning and Placement [Online] https://s2.smu.edu/~manikas/Benchmarks/MCNC_Benchmark_Netlists.html

12) Maegan Tucker, Ellen Novoseller, Preference-Based Learning for Exoskeleton Gait Optimization, 26 Sep 2019, https://arxiv.org/abs/1909.12316

13) https://xueshu.baidu.com/usercenter/paper/show?paperid=1p6n04c01a1h0640nh2606r0rk025842&site=xueshu_se

14) https://zhuanlan.zhihu.com/p/31067515

15) H Kong，T Yan，MDF Wong：Automatic bus planner for dense PCBs. Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE

16) Cheng R, Yan J . On Joint Learning for Solving Placement and Routing in Chip Design[C]，2021.

17) Chang F C, Tseng Y W , Yu Y W , et al. Flexible Multiple-Objective Reinforcement Learning for Chip Placement[J]. arXiv e-prints, 2022

18) Mathur M. Routing and Placement of Macros using Deep Reinforcement Learning[J]. arXiv e-prints, 2022.

19) Mirhoseini, A., Goldie, A., Yazgan, M. et al. A graph placement

methodology for fast chip design. Nature 594, 207–212 (2021). https://doi.org/10.1038/s41586-021-03544-w

20) He Z , Ma Y , Zhang L , et al. Learn to Floorplan through Acquisition of Effective Local Search Heuristics[C]// 2020 IEEE 38th International Conference on Computer Design (ICCD). IEEE, 2020.

21) Qi Xu, Hao Geng, Song Chen, **Bo Yuan**, Cheng Zhuo, Yi Kang, and Xiaoqing Wen, GoodFloorplan: Graph convolutional network and reinforcement learning based floorplanning, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, published online, doi: 10.1109/TCAD.2021.3131550

22) K. Zhu, M. Liu, H. Chen, Z. Zhao, and D. Z. Pan, Exploring logic optimizations with reinforcement learning and graph convolutional network, in ACM/IEEE Workshop on Machine Learning for CAD(MLCAD), 2020, pp. 145–150.

23) A. Goldie and A. Mirhoseini, Placement optimization with deep reinforcement learning, in ACM International Symposium on Physical Design (ISPD), 2020, pp. 3–7.

24) H. Liao, Q. Dong, X. Dong, W. Zhang, W. Zhang, W. Qi, E. Fallon, and L. B. Kara, Attention routing: track-assignment detailed routing using attention-based reinforcement learning, arXiv preprint arXiv:2004.09473, 2020.

25) A. Agnesina, K. Chang and S. K. Lim, VLSI Placement

Parameter Optimization using Deep Reinforcement Learning, IEEE/ACM International Conference On Computer Aided Design (ICCAD), San Diego, CA, USA, 2020, pp. 1-9.

26) Zhang, F. Deng and X. Yang, FPGA Placement Optimization with Deep Reinforcement Learning, 2021 2nd International Conference on Computer Engineering and Intelligent Control (ICCEIC), Chongqing, China, 2021, pp. 73-76.

QA:

1. 赛题要求的模块和区域为什么是多边形

一般模拟 IC 模块都是矩形，但是有的相邻模块彼此端口都连接，模块可以 overlap，为了节省空间，可以看成是一个模块，所以模块需要支持是多边形。本次赛题的模块多边形指的是矩形，正交的 L 型和 T 型，一般来讲，T 型模块较少。

而赛题的背景是在角落区域，已经摆放了其他模块，所以摆放的区域限定是多边形，一般而言，多边形区域可以看成是矩形区域，然后在四个角落上挖去一些矩形，整体摆放区域的顶点数不超过 20 个。

2. 模块为什么需要不同的旋转

由于模块端口之间有连接关系，为了减少线长，模块需要旋转让整体布线的线长较短，同时模块有多边形，旋转才能让整体布局的面积更小

3. 每个模块端口数量及总体的连接数量情况

每个模块的端口数量不超过三个，net（连接文件定义的一条 link）的数量

基本与模块的数量成正比，一般占到模块数量的一半左右，大概率在模块数量的 1/3 到 2 / 3 之间

4. 测试 Case 的情况说明

为了测试算法的普适性，打分 Case 的模块大小和连接关系基本都是不同的，达到测试 Case 多样性的目的

5. 华大提供的评价程序的 Linux 版本要求

目前评价程序支持的 Linux 版本是 CentOS 6.9

6. 传统的启发式/优化算法的运行时间可以满足时间限制的要求么？

对于模块数量非常少，传统启发式/优化算法可以满足要求，但是随着模块逐渐增多，搜索布通率高的布局结果的时间将会急剧增加，因为本身布线评价程序需要一定的运行时间。对于模块越多，传统的启发式/优化算法在限定时间内较难搜索到布通率高，且线长面积指标优秀的结果，因此需要使用 AI 的技术对传统方法速度慢的算法部分进行加速优化，让时间满足限制的要求。而时间的打分采取的是运行时间的相对值，AI 优化的算法部分越多，时间上越有优势。

如果整体算法都是采用 AI 的算法技术，时间上将会比传统启发式/优化算法与 AI 结合的技术更有优势。