【赛题解析】组合逻辑优化与工艺映射的智能流程

一、所需知识背景

1. 专业背景不限。该赛题提供逻辑优化和工艺映射项目的平台，使得参赛者可以着重在智能流程算法的实现，没有专业门槛，对决策、设计空间探索，强化学习等问题感兴趣的同学均可参加；

2. 编程语言不限。在保证可在服务器上运行的情况下，不限制参赛者使用 python、C、C++、shell 等编程语言；

3. 本科、硕士、博士均可参加。

二、赛题背景

组合逻辑优化与工艺映射是数字设计前端的重要流程，组合逻辑的网表，经过对逻辑的化简和再映射为工艺库中的逻辑单元(LUT)组成的逻辑等价的网表，以期达到较优的面积和时延。

三、赛题描述

1. 问题描述

传统的前端综合流程一般是根据工程师的经验，固化一套优化序列和工艺映射来保障逻辑综合阶段生成网表的面积和时延质量。本赛题针对给定的 AIG（And-Inverter Graph）形式的输入组合逻辑电路，要求智能流程算法根据网表的特征，动态地给出合适的逻辑优化算法序列，最终输出功能等价的 LUT（look-up table）网表，并优化逻辑级数/深度和 LUT 数量/面积。

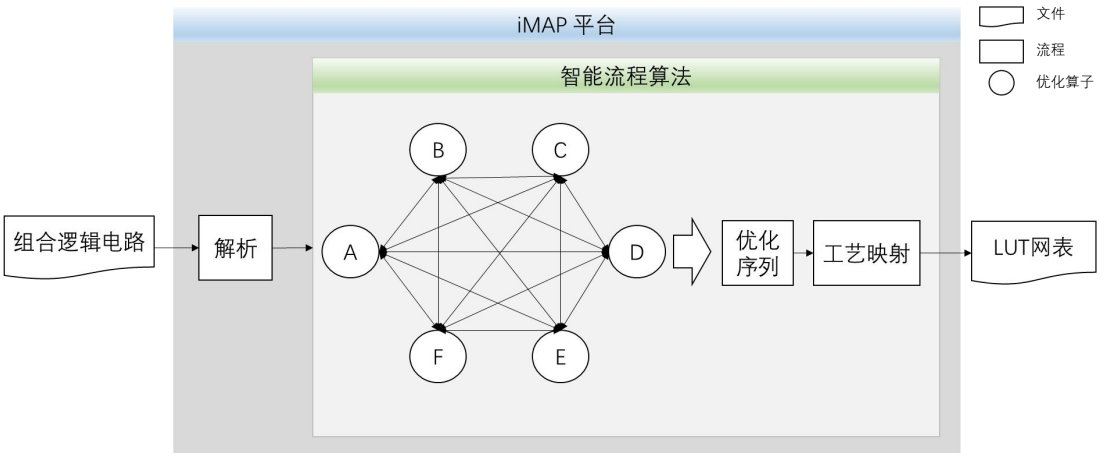

图 1 赛题主体流程

赛题程序的主体流程如图 1 所示，赛题指定编程平台为 iMAP，参赛者可以

依据算法构思和工作量评估，调用这些基础算法组合智能流程，以输出评估结果较优的网表。由于在同一个组合电路中，较优（一般越小越好）的面积和时延往往不可兼得，所以在实际的过程中，输出的网表结果需要在面积和时延进行折衷，从而得到一个较为综合质量的结果。

2. iMAP 平台介绍

```
username@pcl-X11DPi-N-T:<project_path>/bin$ ./imap
imap> read_aiger -f ../../../benchmark/EPFL/arithmetic/adder.aig
imap> print_stats -t 0
Stats of AIG: pis=256, pos=129, area=1020, depth=255
imap> rewrite
imap> balance
imap> refactor
imap> lut_opt
imap> map_fpga
imap> print_stats -t 1
Stats of FPGA: pis=256, pos=129, area=215, depth=75
imap> write_fpga -f adder.fpga.v
```

图 2 iMAP 软件运行图

iMAP 使用 C++编程语言，实现了 rewrite、refactor、balance、map_fpga 等基础逻辑优化以及工艺映射算法。iMAP 平台同时实现了.aig 格式的文件解析和.v 格式的 LUT 网表输出，让参赛者可以专注于智能流程设计。如图 2 所示，在实际使用过程中，参赛者可直接调用给定的优化命令，而不用过于关心内部的实现，并且 iMAP 的运行支持两种方式：command，python 接口。参赛者需按赛题指南提交可执行文件和说明文档。iMAP 平台地址如下：

1）gitee: https://gitee.com/oscc-project/iMAP

2) github: https://github.com/oscc-project/iMAP

四、赛题解析

本赛题的目标是通过实现一种智能流程算法，能够动态地适应不同的输入组合电路，并且针对该电路生成一个优化序列，在经过最终的工艺映射后，能够在综合指标上实现较优的网表面积和时延。因此，该智能流程算法的挑战主要包含以下两方面：(1)从局部和全局的角度来针对当前电路的特征，选择较为合适的当前步骤的优化指令；(2)由于面积和时延往往不可兼得，还需要进行一定的折衷。(3)由于给定时间约束，对参赛者的搜索算法的效率有一定的要求。

五、赛题思路

赛题思路不代表标准的解题步骤，仅供参考，欢迎参赛者在参赛过程中尽情发挥。

1. 思路一：基于分类的算法

   例如，可以通过提取电路的特征，以及该电路在不同优化算子优化后结

果质量，来生成分类任务的数据集；通过神经网路进行分类任务的监督学习生成训练的权重。在实际决策过程中，可以根据当前电路的特征，使用训练后的网络选择合适的优化算子。

2. 思路二：基于贝叶斯优化的多目标优化
   基于贝叶斯优化进行多目标优化是一种常用的方法，可以帮助解决在多个目标函数下找到最优解的问题。这里的挑战是在面积和时延耦合的情况下，目标函数应该如何进行定义。
3. 思路三：基于强化学习的设计空间探索
   该思路主要是如何设置强化学习的状态空间以及动作空间。在该问题下，状态空间可以看作在 step(i)下使用优化算子 action(i)后生成的当前电路的特征，该特征可以是整个电路，也可以是提取的部分特征。动作空间则可以看作优化算子的组合空间。然后通过选择合适的强化学习模型来进行探索和优化。

六、注意事项

1. 输入输出

1）输入文件：每个案例的输入文件为 AIG（And-Inverter Graph）格式的组合逻辑电路。需要注意的是，对于一个 AIG 文件，我们可以通过 iMAP 工具的 API 获取丰富的特征信息，以供参赛者使用；

2）输出文件：对于每个案例的输出文件为智能流程的序列，序列中为 iMAP 所支持的带参数的优化算子。

3）时间约束：数据集分为小，中，大三类数据集，在给定时间的约束下，在服务器上执行参赛者输出的输出文件序列。

2. 自定义特征

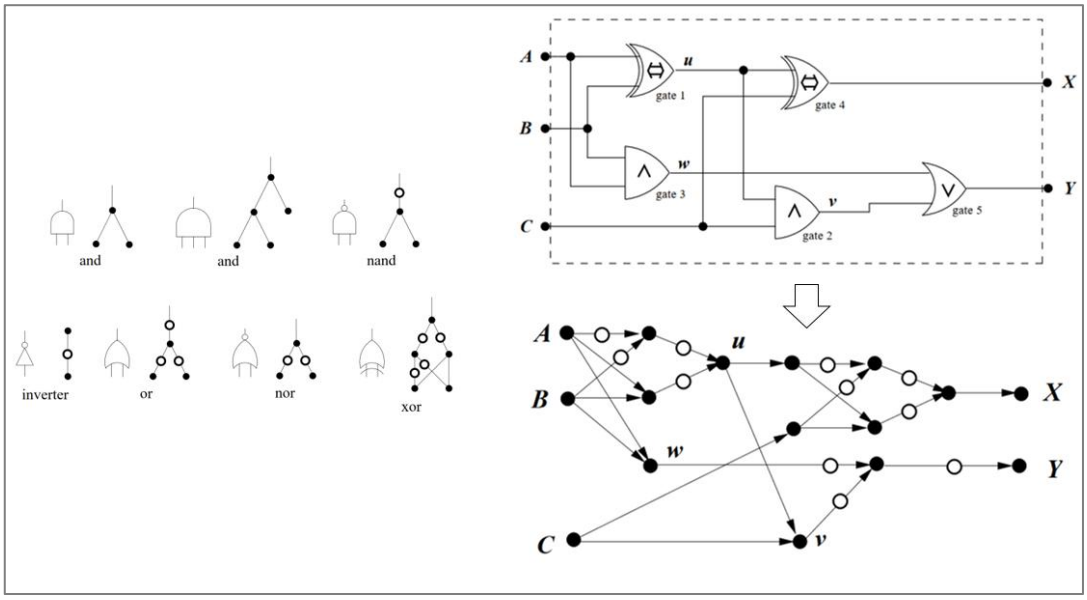

图 3 And-Inverter Graph (AIG)格式

图 3 所示是 AIG 图的结构图。在参赛者实现算法过程中，可以针对算法的需要，通过重写 iMAP 工具内的的 print_stats 命令来获取更为丰富的特征，以供算法需要。目前 print_stats 只提供 AIG 的输入输出数量，节点数量以及最大深度作为参考，还有更为丰富的特征，比如非门的个数，节点度的分布等，子树结构的等。

2. 评分标准

1） 保证逻辑等价。根据参赛者代码执行输出的文件进行组合电路优化以及工艺映射的最终输出的 FPGA Verilog 文件需要和原始输入的 AIG 逻辑等价；

2）在保证 1）的情况下，cost = 0.6*delay+0.4*area，结果质量越高越好，正相关与 cost 越低越好。

参考资料

[1] A. Mishchenko, R. Brayton, S. Jang and V. Kravets, "Delay optimization using SOP balancing"

[2] A. Mishchenko, S. Chatterjee and R. Brayton, "DAG-aware AIG rewriting: a fresh look at combinational logic synthesis"

[3] Brayton A M R. “Scalable logic synthesis using a simple circuit structure”

[4] S. Chatterjee, A. Mishchenko, R. Brayton, X. Wang and T. Kam, "Reducing structural bias in technology mapping"

[5] D. Chen and J. Cong, "DAOmap: a depth-optimal area optimization mapping algorithm for FPGA designs"

[6] Xing Li, Lei Chen, Fan Yang, Mingxuan Yuan, Hongli Yan, and Yupeng Wan, “HIMAP: a heuristic and iterative logic synthesis approach”

[7] Antoine Grosnit, Cedric Malherbe, Rasul Tutunov, Xingchen Wan, Jun Wang, Haitham Bou Ammar, “BOiLS: Bayesian Optimisation for Logic Synthesis”, DATE, 2022.

[8] Cunxi Yu, “FlowTune: Practical Multi-armed Bandits in Boolean Optimization”,ICCAD, 2020.

[9] Abdelrahman Hosny, Soheil Hashemi, Mohamed Shalan, and Sherief Reda, “ DRiLLS: Deep Reinforcement Learning for Logic Synthesis”, ASP-DAC 2020.

[10] C. Yang, Y. Xia, Z. Chu, and X. Zha, “Logic Synthesis Optimization Sequence Tuning Using RL-Based LSTM and Graph Isomorphism Network,” IEEE Trans. Circuits Syst. II Express Briefs, vol. 69, no. 8, pp. 3600–3604, Aug. 2022.

[11] K. Zhu, M. Liu, H. Chen, Z. Zhao, and D. Z. Pan, “Exploring Logic Optimizations with Reinforcement Learning and Graph Convolutional Network,” in Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD, Nov. 2020.