2023（第五届）集成电路 EDA 设计精英挑战赛
赛题指南

版本	时间	修订内容
V0.1	2023-08-18	初版
V0.2	2023-09-05	1. 更新 iMap 平台链接 2. 更新主题流程图 3. 更新基础算法介绍

一、赛题名称

组合逻辑优化与工艺映射的智能流程

二、命题单位

上海安路信息科技有限公司、鹏城实验室、北京大学高能效计算与应用中心

三、赛题 Chair

李兴权（鹏城实验室）

四、赛题描述

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）组成的等价网表，以求达到优化的面积和时序。

现有工具多采用固定的流程完成优化和映射，本赛题针对给定的 AIG（And-Inverter Graph）网表，要求智能流程根据网表特征动态地给出合适的优化算法组合，输出功能等价的 LUT（look-up table）网表，并优化逻辑级数和 LUT 数量。

赛题程序的主体流程如图 1 所示，各个模块详细解释见后续内容。赛题指定编程平台为 iMap（https://gitee.com/oscc-project/iMAP），该平台使用 C++编程语言，实现了 rewrite、refactor、balance、lut-mapping 等基础算法，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（不允许调用第三方工具和算法）。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。

参赛者需提交程序可执行文件，以及必要的文档说明；鼓励提交源代码。

若选择不提交源码，可执行文件需根据输入案例输出 iMap 平台提供的优化命令序列，由赛题方根据产生的命令序列对该案例进行验证。

本赛题限制使用单核单线程。

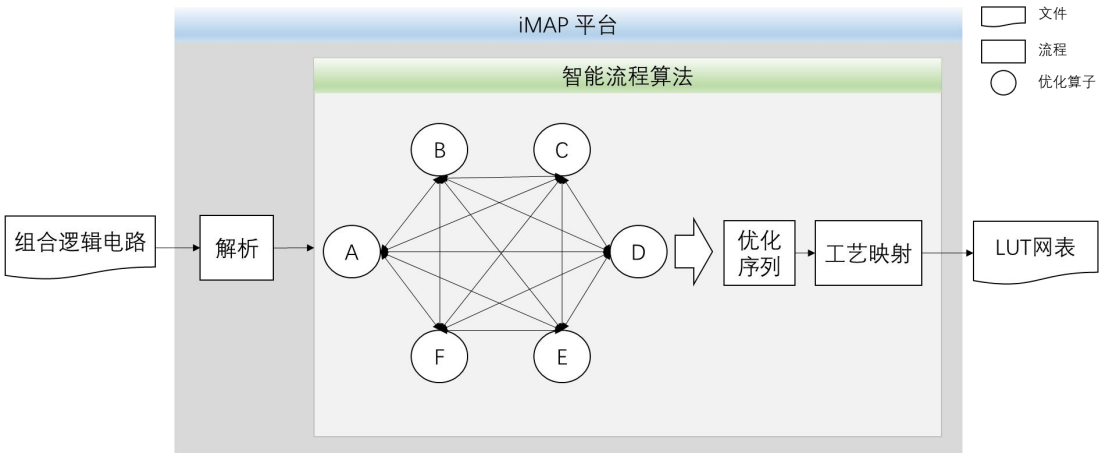

图 1 赛题主体流程

五、输入文件

赛题的输入文件为组合逻辑的.aig 格式(https://fmv.jku.at/aiger/)。一个典型的输入文件如下：

```
// Header
//   M = maximum variable index
//   I = number of inputs
//   L = number of latches
//   O = number of outputs
//   A = number of AND gates

// an OR gate example: o0 = !(!i0 & !i1) = i0 | i1
     aig 3 2 0 1 1
     2                          input 0
     4                          input 1
     7                          output 0     !(!i0 & !i1)
     6 3 5                      AND gate 0 !i0 & !i1
```

本次赛题只针对组合逻辑网表，所以第三项(L) 总是 0。

六、 iMap 基础算法

算法/命令	输入	可选参数	输出	可获得特征
balance	AIG 网表	无	AIG 网表	AIG 网表大小，深度
rewrite	AIG 网表	1.是否零增益替换 2.是否保留深度	AIG 网表	AIG 网表大小，深度
refactor	AIG 网表	1.是否零增益替换 2.是否保留深度 3.最大 cut size 4.最大 cone size	AIG 网表	AIG 网表大小，深度
lut_opt	AIG 网表	1.最大 LUT 输入数量 2.单个节点 cut 数量限制 3.全局面积恢复迭代次数 4.局部面积恢复迭代次数	AIG 网表	AIG 网表大小，深度
map_fpga	AIG 网表	1.单个节点 cut 数量限制 2.全局面积恢复迭代次数 3.局部面积恢复迭代次数 4.支持带 choice 的 LUT mapping	LUT 网表	LUT 网表面积，级数

七、输出文件

输出文件为 LUTs 组成的网表。一个 K 输入的 LUT 本质

上是一个K位地址的单输出RAM，可以通过存储真值表的方式实现任意K输入的布尔逻辑。

一个典型的输出文件如下：

其中 o0 = i0&i1&i2&i3&i4&i5; o1 = i0 | i2 | i4;

```
module top (\i0 , \i1 , \i2 , \i3 , \i4 , \i5 , \o0 , \o1 );
	input \i0  ;
	input \i1  ;
	input \i2  ;
	input \i3  ;
	input \i4  ;
	input \i5  ;
	output \o0  ;
	output \o1  ;
	wire _w9_ ;
	wire _w8_ ;
	LUT6 name0 (
		\i0 ,
		\i1 ,
		\i2 ,
		\i3 ,
		\i4 ,
		\i5 ,
		_w8_
	);
	defparam name0.INIT = 64'h8000000000000000;
	LUT3 name1 (
		\i0 ,
		\i2 ,
		\i4 ,
		_w9_
	);
	defparam name1.INIT = 8'hfe;

	assign \o0  = _w8_ ;
	assign \o1  = _w9_ ;
endmodule;
```

八、参考文献

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

SOP balancing", ICCAD, 2011.

[2] A. Mishchenko, S. Chatterjee and R. Brayton, "DAG-aware AIG rewriting: a fresh look at combinational logic synthesis", DAC, 2006.

[3] Brayton A M R. “Scalable logic synthesis using a simple circuit structure”, IWLS, 2006.

[4] S. Chatterjee, A. Mishchenko, R. Brayton, X. Wang and T. Kam, "Reducing structural bias in technology mapping", ICCAD, 2005.

[5] D. Chen and J. Cong, "DAOmap: a depth-optimal area optimization mapping algorithm for FPGA designs", ICCAD,2004.

[6] Xing Li, Lei Chen, Fan Yang, Mingxuan Yuan, Hongli Yan, and Yupeng Wan, “HIMap: a heuristic and iterative logic synthesis approach”, DAC, 2022.

[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.

九、结果评估

赛题的所有测试案例根据输入规模分为大、中、小三类测试案例。赛题从 3 类测试案例中筛选并提供一部分案例给参赛者评估算法的质量;其余案例只作评分用途,不开放给参赛者。

赛题的每个测试案例包含如下文件:

- .aig 输入文件;
- 质量参考表（定义了该案例的总面积参考值、最大时延参考值和限定运行时间）

对于每个测试案例，有以下几个方面的结果质量评估:

- 通过逻辑等价验证工具，证明输出 LUT 网表和输入 AIG 网表等价（大案例出于验证时间过长的考虑，可以不要求等价验证);
- 输出网表的总面积和最大逻辑级数，两者越小越好，可以结合参考值作评估;
- 算法的运行实际在限定时间范围内。

十、评分方式

每个测试案例有独立的评分，遵循同样的评分标准：

（一） 若案例的逻辑等价性验证失败，或仿真失败，或者超时，则该案例的评分为 n（n 为参赛队伍的数量）。

（二） 案例通过条件 1）的情况下，假设参赛队伍有 n 支，分别按总面积和最大时延的结果质量排序，第 1 名得分 1，第 2 名得分 2，以此类推，最后一名得分 n，然后按 0.6 * 最大时延得分+ 0.4 * 总面积得分的公式计算案例得分。例如某支队伍在某个案例上总面积得分 1，最大时延得分 5，则案例得分为 0.6 * 5+ 0.4 * 1 = 3.4

（三） 案例通过条件（一）的情况下，如果存在结果质量相同的情况，则按分数相同，名次递增的规则评分。例如有 2 支队伍都是总面积结果最佳，则这 2 支队伍在总面积的评分上得分都是 1，第 3 个次优队伍得分 3。

对评分案例集中的每个案例评分后，总得分为单个案例得分的加权求和，其中单个案例的权重系数与该案例的规模和复杂度正相关。总得分越低，排名越高。