



# 第六届“复微杯”全国大学生电子设计大赛

## 支持动态图的 AI 编译器设计

### 软件赛道赛题一

---

# 1 赛题简介

动态的特征在当前的深度学习网络，特别是自然语言处理领域里，有许多体现，比如：动态的控制流，动态的算子参数及动态的数据张量尺寸等。而现有的神经网络编译器，进行编译优化时往往只能针对静态的模型。如何处理动态模型的编译优化，是当前 AI 编译器适应当前 AI 模型发展的重要需求，也是面临的重大挑战。

在深度学习编译器领域，MLIR(Multi-Level Intermediate Representation) 是近年来备受关注的通用 AI 编译器框架，已有许多影响力较大的基于 MLIR 的开源项目，例如 torch-mlir、circt 等。MLIR 定义了多级中间表示的架构，包含 Dialect, Operation, Attribute, Type, Context 等组成部分，同时提供一些通用的程序形成标准库，便于用户基于此架构进行编译器开发，尤其是 AI 编译器。

本赛题主要要求参赛者基于 MLIR 开发一款支持动态图的 AI 编译器，并设计一款支持动态图的 AI 加速器，采用该编译器将一些动态图网络编译到自行设计的加速器上运行。

---

## 2 赛题要求

本赛题将分为两个阶段，初赛和决赛。

### 2.1 初赛

初赛主要要求参赛者基于 MLIR 开发实现支持动态图的 AI 编译器，前端框架为 Pytorch；并设计一款支持动态图的 AI 加速器，加速器支持 FP32 类型的计算(不考虑量化)；采用该编译器完成动态图网络在加速器上的前向推理，功能正确。

**开发实现：**

- 1) 基于 MLIR 实现支持动态图表达的方言（中间层）。
- 2) 将 Pytorch 框架上的动态图模型解析到中间层。
- 3) 利用 MLIR/LLVM 生态，验证转换后的中间层代码的正确性。

**方案及报告：**

针对以上 3 条开发内容，提供完整设计报告，验证及测试报告（以上可以采用已有的开源项目代码，但需要对所用项目深入理解，了解其基本实现原理和代码结构，并在报告中体现）。参赛选手可采用附录中的任意一个或多个模型作为示例以验证编译器的正确性。参赛选手可以在完成上述 3 项的基础上完成以下附加项。如果参赛者在初赛中完成不同的附加项，那么将会在决赛中对参赛选手提出不同的要求，具体要求见下一小节。

- 1)（附加，加分项）设计一款指令驱动的支持动态图的 AI 加速器，该加速器不要求完成电路设计，可以使用软件搭建行为级的仿真模型。
- 2)（附加，加分项）完成动态图到加速器指令的编译过程，即将 MLIR 表示的动态图的中间层一步步下降到加速器的指令序列，并将指令序列在加速器仿真模型上的运行结果和框架上的结果比较，验证编译器和加速器设计的正确性。

## 2.2 决赛

决赛赛题要求参赛选手将编译器进行算子扩展以支持更多的网络模型。决赛时，将提供给参赛选手若干动态图网络模型，选手需要对编译器进行算子扩展来支持这些网络。决赛的具体要求取决于参赛选手初赛时的附加项完成情况，详见下表：

初赛附加项完成情况	决赛要求
未完成任何附加项	1) 将提供的若干动态图网络模型下降到中间层，并利用 MLIR/LLVM 生态验证正确性
只完成了附加项 1	1) 将提供的若干动态图网络模型下降到中间层，并利用 MLIR/LLVM 生态验证正确性 2) 将提供的若干动态图算子通过手工编译得到指令，并利用仿真模型验证正确性
完成全部附加项	1) 将提供的若干动态图网络模型通过编译器以及运行时部署到仿真模型上运行，并验证正确性

## 2.3 输出要求

### (初赛)

- 1) 书面报告。
- 2) 软件工程：
  - 支持动态图的 AI 编译器；
  - 支持动态图的 AI 加速器行为级仿真模型；

### (决赛)

- 1) 答辩 PPT。
- 2) 软件工程：
  - 支持所给动态图网络模型的 AI 编译器
  - 支持所给动态图网络模型的 AI 加速器行为级仿真模型

---

## 2.4 附录

- 1) MLIR 代码链接: <https://github.com/llvm/llvm-project/tree/main/mlir>
- 2) 一些动态图网络模型下载链接:

**drl4vrp:**

<https://github.com/mveres01/pytorch-drl4vrp>

**SCNN:**

<https://github.com/Turoad/lanedet/blob/main/lanedet/models/aggregators/scnn.py>

**DETR:**

<https://github.com/lyuwenyu/RT-DETR>

**SuperGlue:**

<https://github.com/magicLeap/SuperGluePretrainedNetwork>

---

## 3 报告模板

初赛阶段的报告模板至少应该包括以下部分：

- 1) 团队介绍（团队成员与职责分工）
- 2) 整体设计方案，包括设计理念与思路
- 3) 编译器的整体结构，以及各模块的设计介绍
- 4) 加速器的整体结构，以及各模块的设计介绍
- 5) 编译器的可扩展性设计的思路和方法
- 6) 实现过程中的工作
- 7) 最终模型运行的结果

---

## 4 评分标准（初赛）

初赛评分细则如下：

序号	内容	标准	分值
1	基于 MLIR 实现支持动态图表达的方言（中间层）	算子个数；对动态图的支持	20
2	实现 Pytorch 上动态图的解析	功能正确；	20
3	利用 MLIR/LLVM 生态验证中间层的正确性。	功能正确	20
4	实现支持动态图的 AI 加速器行为级仿真模型	功能正确；	30
5	实现动态图的编译和运行过程	优化效果；优化方法；结果分析	30
6	报告	完整性；简洁准确、逻辑清晰；创新性	20

---

## 5 赛事安排

### 5.1 参赛队伍要求

每支参赛队伍控制在 5 人以内。

### 5.2 赛事流程

阶段	时间	内容
初赛	约 2~3 个月	根据本赛题要求完成并提交相应设计和书面报告 确定决赛队伍
培训		对进入决赛的队伍进行相应培训
决赛	约半个月	统一上机决赛
答辩		决赛答辩，确定名次与奖项

注：具体安排以组委会通知为主。