

2025 中国研究生创“芯”大赛 · EDA 精英挑战赛

一、赛题名称

电路系统框图识别与解析

二、命题单位

国家集成电路设计自动化技术创新中心

三、赛题背景

随着电子设计自动化（EDA）技术的飞速发展，电子工程领域的学术论文、技术报告和专利文档正变得越来越数据密集和多媒体化。这些文档不仅包含大量文字描述，还嵌入了丰富的系统框图图片（如电路逻辑图、架构示意图），用于直观展示复杂的设计原理、信号流和控制结构。例如，在一篇关于集成电路设计的论文中，系统框图可以占到整篇内容篇幅的 30%以上，它并非简单的插图，而是承载了关键的系统级信息，如模块互连、时序控制和性能指标。然而，现有的大模型（如大语言模型 LLM 或视觉模型）在解析这些多模态数据时表现不足：大语言模型虽能处理文本信息，但对图片内容视而不见；纯视觉模型虽可识别图像元素，却无法结合上下文文本深度理解逻辑语义；这种割裂导致模型在自动摘要、设计验证或知识检索等应用中出现高错误率，如错误解读框图路径或忽略关键模块依赖。据统计，人工标注师在解析此类文档时消耗时间长（平均每页需 15-20 分钟），且由于专业门槛高，易出现主观偏差。如果大模型能实现高效的图文协同理解，它可大幅加速 EDA 设计周期，例如自动生成设计报告、辅助电路优化或支持智能教育工具，最终推动电子创新。

该赛题的核心挑战在于如何让模型不仅“看图识物”，还能“融图入文”：既要学习系统框图的视觉特征（如模块布局、连线类型），这需要多模态模型在预训练阶

段吸收海量文档数据，通过注意力机制融合图文表示，并在下游任务中推广到多样场景。又要解析系统框图的逻辑特征（如组件功能解析，连接关系判定），这需要多模态模型对系统框图进行分析，用于回答所给出的问题。然而，当前开源模型在此领域数据稀缺（缺乏标注的多模态文档库）、架构不成熟（融合层效率低），导致泛化能力弱、多模态问题推理能力差。本赛题旨在填补这一缺口，激励参赛者开发创新模型，促进 AI 在 EDA 中的实际落地。最终，赛题的成功将为工业界和学术界提供开源工具，减少人工成本，提升设计可靠性和创新速度，并为 AI 通用图文理解能力开辟新径。

四、赛题描述

在集成电路(IC)设计的宏伟蓝图中，系统级框图作为一种高度抽象的视觉语言，是架构师思想的结晶，承载着整个芯片的核心功能、模块划分与数据流向。然而，随着设计日益复杂，海量的存量设计文档（尤其是图像格式的框图）形成了一座巨大的“信息孤岛”，其分析、验证、迁移和复用高度依赖人工，成为制约设计迭代效率和智能化的瓶颈。

本赛题聚焦于人工智能与 IC 设计的交叉前沿领域，致力于运用多模态大模型(VLMs)攻克一项关键挑战：从系统级框图图像中，自动化地解析其拓扑结构与逻辑关系，实现非结构化视觉信息到结构化设计知识的精准转换。

本赛题要求参赛者构建一个系统框图识别算法，该算法需具备以下核心能力：

跨模态语义认知 (Cross-modal Semantic Perception) : 模型不仅要“看见”框图中的视觉元素（如矩形、菱形、文本标签），更要“理解”它们所代表的 IC 设计中的具体含义，如“处理器核心”、“内存控制器”、“总线接口”等异构功能组件。

复杂拓扑结构推理 (Complex Topology Reasoning) : 模型需要超越简单的线条检测，精确解析组件之间错综复杂的连接关系。这包括识别点对点连接、多路复用、总

线结构以及信号流向等，并能处理线条交叉、遮挡、断点等视觉噪声。

逻辑拓扑的结构化重建（Structured Reconstruction of Logical Topology）：最终目标并非零散的识别框和线段列表，而是要将识别出的所有组件（作为逻辑节点）和连接关系（作为逻辑边）进行系统性整合，重建为一个完整且精确的结构化数据模型（Structured Data Model）。该模型需清晰地定义每一个组件的属性（如功能类别、文本标识）以及组件间的连接拓扑关系。此输出必须与基准真值（Ground Truth）标签文件中的数据，在组件和连接关系两个维度上，均实现严格的、无歧义的一一对应（strict and unambiguous one-to-one mapping），确保解析结果的绝对保真度。

系统框图逻辑解析（Logical Analysis of System Block Diagram）：目标是全面分析与推理系统框图中的图文信息，通过整合视觉元素与文本描述，精准理解框图所表达的设计逻辑与功能意图。最终，模型需能够基于这些信息，准确回答与框图相关的逻辑推理问题，验证其对系统框图的整体理解能力。

赛题属于集成电路领域。具体描述为给定一组不同结构的逻辑电路系统框图和描述系统框图中组件与组件之间连接关系的标签，如何使用 CV 算法，同时结合多模态大模型将尽可能多的组件和连接关系识别出来，并与标签文件中的数据一一匹配，同时完成所给出的逻辑分析问题。

赛题数据：

赛题为参赛队伍提供 1000 张系统框图，其中 20 张系统框图含有标签数据，作为测试集；对于未标注的训练数据，可利用现有的多模态大模型得到一些初步的标注。同时，允许参赛队伍自行搜集其它数据进行模型的训练。

输入：

一个电路系统框图，如图 1

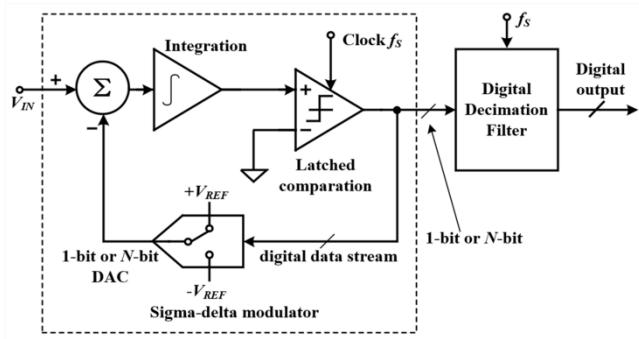


图 1

这是一个标准的电路系统框图，赛题目标为利用 CV 算法，同时结合多模态大模型对图形进行识别，以生成尽可能多且精确的组件信息，并完成赛题所指定的任务。

输出：

任务一（60%）：

在任务一部分，本赛题将模型的输出分为三个阶段：组件提取、组件输入输出端口数量匹配、连接关系提取。

(1) 组件提取：识别出图片中的组件名称及位置信息，按照规定格式给出。

组件定义：电路图中的元器组件，在图中显示为图标。

输出格式：{[Component: "A", pos: ["x1", "y1", "x2", "y2"]]}

其中，Component 为组件的名称，严格按照图中的组件昵称来命名，若无名称则使用组件的类型进行命名；pos 中的["x1", "y1", "x2", "y2"]分别为图形 IoU 计算时 box 的[左上角的 x, 左上角的 y, 右下角的 x, 右下角的 y] (x,y 均取相对值)。

(2) 组件输入输出端口数量提取：对每个组件进行分析，统计每个组件的输入何输出端口数量，并生成一个序列。

组件输入输出端口定义：用于接收指向目标组件和目标组件发送指向其他组件的连接关系的平台，如用于→，--的端口，指向目标组件的平台为输入端口，从目标组件发送给其他组件的平台为输出端口。

输出格式：例如图形 Σ 含有两个输入端口，一个输出端口，那么它的端口信息则为 I_O:{Component: Σ ,input:2,output:1}。

(3) 连接关系提取：识别组件与组件之间的连接关系，并按照指定格式给出。

连接关系定义：组件和组件联系的箭头或者其他指向性符号。

输出格式：

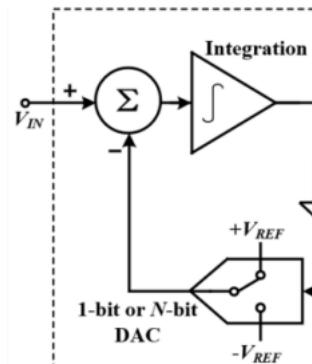


图 2

使用大模型生成组件连接关系，假定有组件连接关系“A→B， B→C”，那么组件 B 的连接关系表达形式为：[input: ["A"] ; output: ["C"]]，例如在图 2 中，对于组件 Σ 有连接关系[input: ["V IN","1-bit or N-bit DAC"];output["Integration"]]。

(4) 最终结果：

最终结果将按照 Json 格式输出。

(注：module 指的是每一张参与模型识别的系统框图，一个图片则用一个 module 来记录内部组件信息。Component 为单个组件的名称，Pos 是组件进行 IOU 计算时的 BOX 坐标，I_O 是组件的输入输出端口，Connection 是组件与其他组件之间的连接关系。)

```
OutputJson{
{
  "image1": [
    {
      "Component": "B",
      "Pos": ["x1", "y1", "x2", "y2"],
      "I_O": {
```

```

    "input": 1,
    "output": 1
  },
  "Connection": {
    "input": ["A"],
    "output": ["C"]
  }
},
{
  "Component": "E",
  "Pos": ["x1", "y1", "x2", "y2"],
  "I_O": {
    "input": 2,
    "output": 2
  },
  "Connection": {
    "input": ["C", "D"],
    "output": ["F", "G"]
  }
}
],
"image2": [
  {
    "Component": "B2",
    "Pos": ["x1", "y1", "x2", "y2"],
    "I_O": {
      "input": 1,
      "output": 1
    },
    "Connection": {
      "input": ["A2"],
      "output": ["C2"]
    }
  },
  {
    "Component": "E2",
    "Pos": ["x1", "y1", "x2", "y2"],
    "I_O": {
      "input": 2,
      "output": 2
    }
  }
]

```

```
    },
    "Connection": {
        "input": ["C2", "D2"],
        "output": ["F2", "G2"]
    }
},
],
}
```

任务二（40%）：

在任务二部分，本赛题要求参赛队伍结合多模态大模型，对赛题给出的模拟电路问题进行推理回答，推理出尽可能多且正确的 QA 问答对。

其中，在选择题部分，模型需要预测出一个选项，当预测选项与答案选项一一对应时即可获得分数；在填空题部分，只有当模型预测的结果与正确答案完全匹配时，才可获得相应的分数。

QA 对生成：

本部分将给出模拟电路常识/模拟电路分析的相关问题，要求多模态大模型对问题进行分析，给出预测答案，并与正确答案进行比对，以生成准确率。具体生成格式如下：

例：

对于图 3，有相关问题如下：

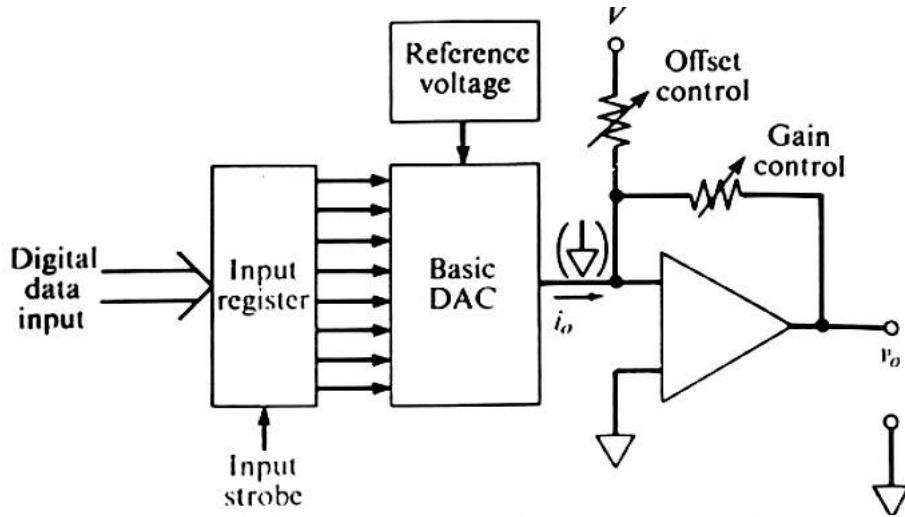


图 3

选择题：

1. 考虑图中 Basic DAC 的电流输出经由可调反馈电阻 R_f (Gain control) 输入运算放大器反相端，且反相输入节点存在等效电容 C_{in} 。以下叙述中，哪一项正确描述了闭环带宽 f_c 随增益控制（反馈电阻值）的变化关系？
 - A. 无论 R_f 取何值，闭环带宽均保持不变，因为 Op Amp 的开环带宽主导响应。
 - B. 随着 R_f 增大，闭环带宽减小，大致满足 $f_c \approx \frac{1}{2\pi R_f C_{in}}$ 的反比关系。
 - C. 随着 R_f 增大，闭环带宽增大，因为更高的增益有利于高频响应。
 - D. 随着 R_f 增大，闭环带宽先增大后减小，呈非单调变化，这是由于增益控制与 Offset control 耦合造成的。

答案：B

若模型预测答案为 C，则视为匹配错误，无法获得相应分数。

2. 基于图中所示电路，假设 Basic DAC 的输出电流为 I_o ，偏置电位器 Offset control 的端到端电阻中，连接至参考电压 V_+ 与运放反相输入节点之间的等效电阻为 R_{off} ，反馈电阻 Gain control 为 R_f ，则输出电压 v_o 与 I_o 、 V_+ 、 R_f 和 R_{off} 之间的关系，下列哪一

项正确？

A. $v_o = -I_o R_f + \frac{V_+}{R_{off}} R_f$

B. $v_o = -I_o R_f - \frac{V_+}{R_{off}} R_f$

C. $v_o = I_o R_f - \frac{V_+}{R_{off}} R_f$

D. $v_o = I_o R_f + \frac{V_+}{R_{off}} R_f$

答案：B

若模型预测答案为 B，则视为匹配正确，可获得相应分数。

A. $v_o = -I_o R_f + \frac{V_+}{R_{off}} R_f$

B. $v_o = -I_o R_f - \frac{V_+}{R_{off}} R_f$

C. $v_o = I_o R_f - \frac{V_+}{R_{off}} R_f$

D. $v_o = I_o R_f + \frac{V_+}{R_{off}} R_f$

填空题：

考虑图中 Basic DAC 的电流输出经由可调反馈电阻 R_f (Gain control) 输入运算放大器反相端，且反相输入节点存在等效电容 C_{in} 。

请回答：随着 R_f _____，闭环带宽减小。

答案：增大

若模型预测结果为增大，则视为匹配成功，可获得相应分数。

五、评分标准

(1) 约束条件

如参赛队伍使用 VLM, 要求参赛队伍使用的推理大模型类型为:Qwen2.5-VL-3B, Adapter 参数量应少于 1B, 总计算模型参数应小于 4B。使用其他大模型参与将视为无效参赛。注意: 该要求仅限于测试过程。

(2) 评分方式

设置几个参数, 用于计算 F1 分数:

TP (True Positive) : 被正确预测的正例。即该数据的真实值为正例, 预测值也为正例的情况;

TN (True Negative) : 被正确预测的反例。即该数据的真实值为反例, 预测值也为反例的情况;

FP (False Positive) : 被错误预测的正例。即该数据的真实值为反例, 但被错误预测成了正例的情况;

FN (False Negative) : 被错误预测的反例。即该数据的真实值为正例, 但被错误预测成了反例的情况。

根据赛题设定的输出部分, 本赛题以加权平均的形式为每位参赛队伍赋分。假定 S1,S2,S3 分别为参赛者在输出(1),(2),(3)中的得分, 其中, S1,S2,S3 的分数设定不同的得分点, 每正确匹配一次数据将得到相应的分数。

任务一 (60%) :

组件提取评分标准 S1 :

匹配 `box:{[Component: ,pos:],[...]}` 中的位置信息, 根据目标检测参数 IoU (交并比) 来进行计算, 当 $\text{IoU} \geq 0.5$ 时视为位置匹配正确, 如果存在多个 $\text{IoU} \geq 0.5$ 的 box,

则取 IOU 最高的组件作为匹配组件，即可得到相应的分数。

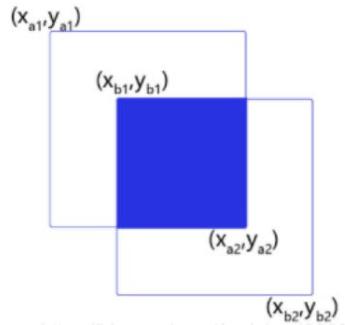


图 4 IOU 计算方式

相交部分左上角坐标为：

$$x_1 = \max(x_{a1}, x_{b1}), \quad y_1 = \max(y_{a1}, y_{b1})$$

相交部分右下角坐标为：

$$x_2 = \min(x_{a2}, x_{b2}), \quad y_2 = \min(y_{a2}, y_{b2})$$

交集面积计算：

$$\text{intersection} = \max(x_2 - x_1 + 1.0, 0) \cdot \max(y_2 - y_1 + 1.0, 0)$$

两个框的面积：

$$S_A = (x_{a2} - x_{a1} + 1.0) \cdot (y_{a2} - y_{a1} + 1.0)$$

$$S_B = (x_{b2} - x_{b1} + 1.0) \cdot (y_{b2} - y_{b1} + 1.0)$$

并集面积：

$$\text{union} = S_A + S_B - \text{intersection}$$

IoU 计算：

$$\text{IoU} = \frac{\text{intersection}}{\text{union}}$$

根据前文提到的参数可以得到：

TP：正确匹配的位置数量 FP:Label 中的位置数量 FN:模型生成的位置数量

则有计算公式为：

精准度：

$$\text{Precision} = \frac{TP}{TP + FP}$$

召回率：

$$\text{Recall} = \frac{TP}{TP + FN}$$

位置匹配评分：

$$S_P = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}$$

完成对 SP 进行的计算后，模型需要生成模型名称，并使用文本编辑距离或词向量计算相似度的方法来对组件名称进行匹配。对于图中有明显名称的组件，要求识别到组件的名称并完成正确匹配，对于没有明显名称的组件，要求按照组件的类型为组件命名，同时正确匹配到相应组件。赛题设置固定阈值，相似度大于阈值则视作正确匹配。

对于模型生成的结果可以分为：

TP：正确匹配的名称数量 FP:Label 中的名称数量 FN:模型生成的名称数量

则有计算公式为：

精准度：

$$\text{Precision} = \frac{TP}{TP + FP}$$

召回率：

$$\text{Recall} = \frac{TP}{TP + FN}$$

名称匹配评分：

$$S_N = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}$$

那么组件提取的评分 S_1 有计算公式如下：

$$S_1 = \frac{(S_N + S_P)}{2}$$

组件输入输出端口数量匹配评分标准 S_2 :

匹配 I_O: {[Component: ,input: ,output: ,],[...]} 中，每个组件的端口数量，当输入端口与输出端口均匹配正确时，即可获得相应分数。（注：当输入端口和输出端口只有一端匹配正确，另一端匹配错误时，无法获得相应分数）

根据前文提到的参数可以得到：

TP: 正确匹配端口的组件数量 FP:Label 中的组件数量 FN:模型生成的匹配数量。

则有计算公式为：

精准度：

$$\text{Precision} = \frac{TP}{TP + FP}$$

召回率：

$$\text{Recall} = \frac{TP}{TP + FN}$$

输入输出端口数量匹配评分 S_2 有计算公式如下：

$$S_2 = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}$$

连接关系提取评分标准 S_3 :

匹配每条连接关系对，如 E 的连接关系对 Connection[input:["C","D"], output:["F","G"]], 每正确匹配一对连接关系对，得到相应的 Case 得分。

根据前文提到的参数可以得到：

TP：正确匹配的数量 FP:Label 中的数量 FN:模型生成的匹配数量。

则有计算公式为：

精准度：

$$\text{Precision} = \frac{TP}{TP + FP}$$

召回率：

$$\text{Recall} = \frac{TP}{TP + FN}$$

输入输出端口数量匹配评分 S_3 有计算公式如下：

$$S_3 = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}$$

对任务一的三部分评分标准进行计算后，将会对任务一总体评分标准进行计算。

任务一总体评分标准 Score_1 :

$$\text{Score}_1 = S_1 \times 40 + S_2 \times 20 + S_3 \times 40$$

$$\text{Score}_1 = S_1 \times 40 + S_2 \times 20 + S_3 \times 40$$

任务二（40%）：

QA 问答对生成：

计算 QA 问答对匹配的准确率，作为该部分的评分标准。

设置 T 为正确匹配，N 为错误匹配，准确率为 PQ，则有该部分的评分标准公式为：

准确率：

$$P_Q = \frac{T}{T + N}$$

任务二得分：

$$\text{Score}_2 = P_Q \times 100$$

模型计算总分评分标准 Score：

在计算总分的过程中，为防止极端分数造成各参赛队伍排名差距过大，赛题使用 Min-Max 归一化，将分数映射在[0,1]区间之中，其计算公式为：

$$\text{Normalized}_x = \frac{x - \min(A)}{\max(A) - \min(A)}$$

其中，x 为参赛队伍在任务 x 所得的分数，如 x1 为该参赛队伍在任务一中获得的分数。 $\min(A)$ 为所有参赛队伍中分数最低的值， $\max(A)$ 为所有参赛队伍中分数最高的值。

对于总分的任务评分要求，有相对评分 $\text{Nomalized}_{x1}, \text{Nomalized}_{x2}$ ，分别对应参赛队伍在任务一和任务二中获得的相对评分。

那么评价总分的计算公式为：

$$\text{Score} = \text{Nomalized}_{x1} \times 60 + \text{Nomalized}_{x2} \times 40$$

模型计算时间要求：

模型的计算时间将会作为赛题评分的加分项，对于计算时间，有以下评分方式：

赛题按照 Benchmark 评测平均时间 AVG 来进行计算

$$AVG = \frac{\text{sum}(\text{runtime}(\text{Benchmark1}) + \text{runtime}(\text{Benchmark2}) + \text{runtime}(\text{Benchmark3}) + \dots)}{\text{count}(\text{Benchmark})}$$

评分专家对 AVG 的大小进行分析，根据表 1 给出参赛队伍在时间层面获得的分数

表 1 计算时间评分表

AVG 队伍排名	得分情况
前 10%	5 分
前 30%	3 分
前 50%	1 分
其他	0 分

例如，在表 2 中，有：

表 2

模型计算时间 /s	Team1	Team2	Team3	Team4	Team5
Benchmark1	40	50	20	80	120
Benchmark2	1780	6800	5000	3000	2300
Benchmark3	2000	3000	500	1200	5000
Benchmark4	3000	4000	2000	5000	3000
AVG	1705	3462.5	1880	2320	2605

AVG 可根据计算公式得出结果如下：

$$AVG_1 = (40 + 1780 + 2000 + 3000) / 4 = 1705s$$

$$AVG_2 = (50 + 6800 + 3000 + 4000) / 4 = 3462.5s$$

$$AVG_3 = (20 + 5000 + 500 + 2000) / 4 = 1880s$$

$$\text{AVG}_4 = (80 + 3000 + 1200 + 5000 + 2320) / 4 = 2320 \text{ s}$$

$$\text{AVG}_5 = (120 + 2300 + 5000 + 3000) / 4 = 2605 \text{ s}$$

对于 Team 1~5，有 AVG 的评分顺序为 Team 1 < Team 3 < Team 4 < Team 5 < Team 2, 那么 Team1 加 5 分，Team 3, 4 加 1 分，Team 5, 2 不加分。

模型 Case 测评：

对于模型的评测 Case，赛题设置为 Public Case 和 Hidden Case 两部分，参赛队伍可以对 Public Case 进行评测来判断模型的性能是否符合赛题要求；在参赛队伍提交模型代码后，组委会将使用 Hidden Case 对模型进行评测，进一步分析各参赛队伍的模型性能。Public Case 与 Hidden Case 的分布比例如表 3：

表 3 模型评测 Case 分布表

类型	Public Case	Hidden Case
比例	60%	40%

六、参赛要求

要求学生具有一定的 Python 语言编程能力，具有一定的逻辑电路，计算机基础。
主要面向计算机、微电子、数学专业的学生，但不限于这些专业。

七、参考文献

- 学生可以借助以下参考文献查阅已有的一些研究，对现有模型进行优化。
- 以下书目涉及集成电路知识及多模态大模型：
- [1] Y. Huang et al., “LayoutLMv3: Unified Text and Image Masking for Document AI,” in Proc. ACM Multimedia, 2022, pp. 1–9.
 - [2] J. Li et al., “Qwen-VL: A Frontier Large Vision-Language Model with Versatile Abilities,” arXiv:2308.12966, 2023.

- [3] Chen, T., et al. “Visual Instruction Tuning: Towards General-Purpose Multimodal LLMs.” NeurIPS 2023.
- [4] Li, J., et al. “Blip-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders.” ICML 2023.
- [5] Liu, S., et al. “LLaVA-OneVision: Easy Visual Task Transfer with Only One Vision-Language Model.” arXiv:2408.03326, 2024.
- [6] Deng, L., et al. “Multimodal Document Intelligence: Models, Benchmarks and Challenges.” ACM Computing Surveys, 2024.
- [7] LayoutLLM-3B: Lightweight Multimodal Transformer for Technical Diagram Understanding arXiv:2405.13214, 2024
- [8] L. Wang et al., “EDA in the Era of Large AI Models: Opportunities and Challenges,” IEEE Trans. CAD, early access, 2024.
- [9] X. Dong et al., “Towards Efficient Large Multimodal Model Serving: A System Perspective,” arXiv:2402.14818, 2024.
- [10] R. Kandur, “Multimodal Large Language Models: Architectures, Challenges and Future Directions,” Int. J. Inf. Technol. Manag. Inf. Syst., vol. 16, no. 1, pp. 430–441, 2025.
- [11] S. Bai et al., “Qwen2.5-VL Technical Report,” arXiv preprint arXiv:2502.13923, Feb. 2025.

*本赛题指南未尽问题，见赛题 O&A 文件

2025 China Postgraduate IC Innovation Competition • EDA Elite Challenge Contest

1. Problem

Identification and analysis of Circuit System Block diagrams

2. Company

National Center of Technology Innovation for EDA

3. Problem Background

With the rapid development of electronic design automation (EDA) technology, academic papers, technical reports and patent documents in the field of electronic engineering are becoming increasingly data-intensive and multimedia. These documents not only contain a large amount of textual descriptions but also embed a wealth of system block diagram images (such as circuit logic diagrams and architecture diagrams), which are used to visually present complex design principles, signal flows and control structures. For instance, in a paper on integrated circuit design, the system block diagram can account for over 30% of the entire content. It is not merely an illustration but carries crucial system-level information such as module interconnection, timing control, and performance metrics. However, existing large models (such as large language models (LLMs) or visual models) perform poorly in parsing these multimodal data: although large language models can handle text information, they turn a blind eye to the content of images; Although pure visual models can recognize image elements, they are unable to deeply understand logical semantics in combination with the context text. This fragmentation leads to a high error rate

of the model in applications such as automatic summarization, design verification, or knowledge retrieval, such as misinterpreting block diagram paths or ignoring key module dependencies. According to statistics, manual annotators take a long time to parse such documents (averaging 15 to 20 minutes per page), and due to the high professional threshold, subjective deviations are prone to occur. If large models can achieve efficient graphic and text collaborative understanding, they can significantly accelerate the EDA design cycle, such as automatically generating design reports, optimizing auxiliary circuits, or supporting intelligent educational tools, ultimately promoting electronic innovation.

The core challenge of this competition question lies in how to enable the model not only to "recognize objects by looking at pictures", but also to "integrate pictures into text": it needs to learn the visual features of the system block diagram (such as module layout and connection type), which requires the multimodal model to absorb a large amount of document data during the pre-training stage, fuse the graphic and text representation through the attention mechanism, and then extend it to various scenarios in downstream tasks. It is also necessary to parse the logical features of the system block diagram (such as component function analysis and connection relationship determination), which requires a multimodal model to analyze the system block diagram to answer the given questions. However, the current open-source models are short of data in this field (lack of labeled multimodal document libraries) and have immature architectures (low efficiency of the fusion layer), resulting in weak generalization ability and poor reasoning ability for multimodal problems. The aim of this competition is to fill this gap, encourage participants to develop innovative models, and promote the practical application of AI in EDA.

Ultimately, the success of the competition will provide open-source tools for both the industrial and academic sectors, reduce labor costs, enhance design reliability and innovation speed, and open up new paths for AI's general graphic and text understanding capabilities.

4. Competition Problem Description

In the grand blueprint of integrated circuit (IC) design, the system-level block diagram, as a highly abstract visual language, is the crystallization of the architect's thoughts, carrying the core functions, module division and data flow of the entire chip. However, as design becomes increasingly complex, a vast amount of existing design documents (especially block diagrams in image format) have formed a huge "information island", whose analysis, verification, migration and reuse are highly dependent on manual labor, becoming a bottleneck restricting the efficiency and intelligence of design iteration.

This competition topic focuses on the cutting-edge intersection of artificial intelligence and IC design, aiming to apply multimodal large models (VLMs) to overcome a key challenge: automatically parsing the topological structure and logical relationship from system-level block diagram images, and precisely converting unstructured visual information into structured design knowledge.

This competition question requires participants to build a system block diagram recognition algorithm, which should possess the following core capabilities:

Cross-modal Semantic Perception: The model not only needs to "see" the visual elements in the block diagram (such as rectangles, rhombuses, and text labels), but also "understand" the specific meanings they represent in IC design, such as heterogeneous functional

components like "processor cores", "memory controllers", and "bus interfaces".

Complex Topology Reasoning: The model needs to go beyond simple line detection and precisely analyze the intricate connection relationships among components. This includes identifying point-to-point connections, multiplexing, bus structures, and signal flow directions, etc., and is capable of handling visual noise such as line crossings, occlusions, and breakpoints.

Structured Reconstruction of Logical Topology The ultimate goal is not scattered recognition boxes and line segment lists, but to systematically integrate all the identified components (as logical nodes) and connection relationships (as logical edges) and reconstruct them into a complete and accurate Structured Data Model. This model needs to clearly define the attributes of each component (such as functional categories and text identifiers) as well as the connection topological relationships between components. This output must achieve strict and unambiguous one-to-one mapping (strict and unambiguous one-to-one mapping) with the data in the Ground Truth tag file in both component and connection dimensions to ensure the absolute fidelity of the parsing result.

Logical Analysis of System Block Diagram: The goal is to comprehensively analyze and reason about the graphic and textual information in the system block diagram. By integrating visual elements and text descriptions, accurately understand the design logic and functional intentions expressed by the block diagram. Ultimately, the model needs to be capable of accurately answering logical reasoning questions related to the block diagram based on this information, verifying its overall understanding of the system block diagram. The competition topic belongs to the field of integrated circuits. Specifically, given a set of

block diagrams of logic circuit systems with different structures and labels that describe the connection relationships between components in the system block diagrams, how to use the CV algorithm and combine it with a multimodal large model to identify as many components and connection relationships as possible, and match them one by one with the data in the label file, while completing the given logic analysis problem.

Competition question data:

The competition question requires each participating team to provide 1,000 system block diagrams, among which 20 system block diagrams contain labeled data, serving as the test set. For unlabeled training data, some preliminary annotations can be obtained by using existing multimodal large models. At the same time, the participating teams are allowed to collect other data on their own for model training.

Input:

A block diagram of a circuit system, as shown in Figure 1:

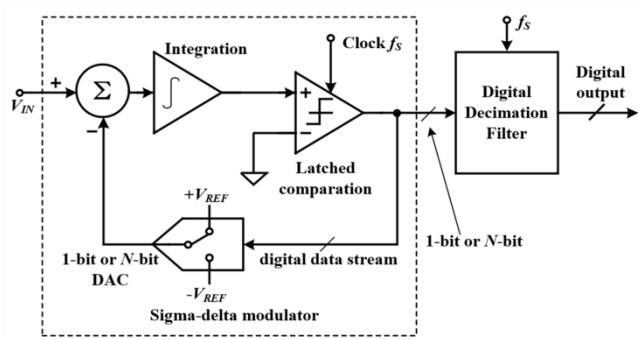


Figure 1

This is a standard block diagram of a circuit system. The objective of the competition question is to use CV algorithms and combine them with multimodal large models to recognize the graphics, in order to generate as much and as accurate component information as possible and complete the tasks specified in the competition question.

Output:

Task One (60%) :

In the task section, this competition question divides the model output into three stages: component extraction, matching the number of input and output ports of components, and connection relationship extraction.

(1) Component extraction: Identify the component names and position information in the image and provide them in the prescribed format.

Component definition: The element component in the circuit diagram, which is displayed as an icon in the diagram.

Output format: {[Component: "A",pos: ["x1","y1","x2","y2"]]}

Among them, "Component" is the name of the component. It should be named strictly according to the nickname of the component in the figure. If there is no name, the type of the component should be used for naming. In pos, ["x1","y1","x2","y2"] respectively represent the [x in the upper left corner,y in the upper left corner, x in the lower right corner,y in the lower right corner] of the box when calculating the graphic IoU (both x and y take relative values).

(2) Extraction of the number of input and output ports of components: Analyze each component, count the number of input and output ports of each component, and generate a sequence.

Component input and output port definition: It is a platform used to receive connection relationships pointing to the target component and for the target component to send connection relationships to other components. For example, the ports used for → and --, the

platform pointing to the target component is the input port, and the platform sending from the target component to other components is the output port.

For instance, if a graphic  contains two input ports and one output port, then its port information would be `I_O:{Component:Σ,input:2,output:1}`.

(3) Connection relationship extraction: Identify the connection relationships between components and provide them in the specified format.

Connection relationship definition: Arrows or other directional symbols indicating the connection between components.

Output format:

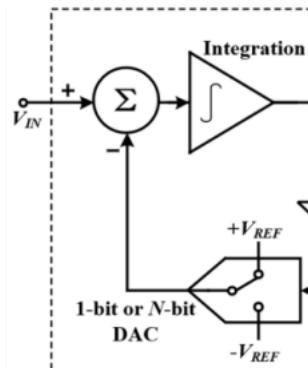


Figure 2

Using A large model to generate component connection relationships, assuming there are component connection relationships " $A \rightarrow B$, $B \rightarrow C$ ", then the connection relationship expression form of component B is: [input: ["A"]; output: ["C"]]. For example, in Figure 2, for component Σ , there is a connection relationship [input: ["V IN","1-bit or N-bit DAC"]; output["Integration"]].

(4) Final result:

The final result will be output in Json format.

Note: "module" refers to each system block diagram involved in model recognition. Each

image uses a module to record the information of its internal components. "Component" is the name of a single component, "Pos" is the BOX coordinate when the component performs IOU calculation, "I_O" is the input and output port of the component, and "Connection" is the connection relationship between the component and other components.

```

OutputJson{
{
  "image1": [
    {
      "Component": "B",
      "Pos": ["x1", "y1", "x2", "y2"],
      "I_O": {
        "input": 1,
        "output": 1
      },
      "Connection": {
        "input": ["A"],
        "output": ["C"]
      }
    },
    {
      "Component": "E",
      "Pos": ["x1", "y1", "x2", "y2"],
      "I_O": {
        "input": 2,
        "output": 2
      },
      "Connection": {
        "input": ["C", "D"],
        "output": ["F", "G"]
      }
    }
  ],
  "image2": [
    {
      "Component": "B2",
      "Pos": ["x1", "y1", "x2", "y2"],
      "I_O": {
        "input": 3,
        "output": 3
      }
    }
  ]
}

```

```

    "input": 1,
    "output": 1
  },
  "Connection": {
    "input": ["A2"],
    "output": ["C2"]
  }
},
{
  "Component": "E2",
  "Pos": ["x1", "y1", "x2", "y2"],
  "I_O": {
    "input": 2,
    "output": 2
  },
  "Connection": {
    "input": ["C2", "D2"],
    "output": ["F2", "G2"]
  }
}
],
}

```

Task Two (40%) :

In Task Two, this competition question requires the participating teams to combine the multimodal large model to reason and answer the analog circuit problems given in the competition question, and to reason out as many correct QA pairs as possible.

Among them, in the multiple-choice section, the model needs to predict an option. When the predicted option corresponds one-to-one with the answer option, points can be obtained.

In the fill-in-the-blank section, only when the predicted result of the model exactly matches the correct answer can the corresponding score be obtained.

QA for generation:

This section will present common knowledge about analog circuits and related issues in

analog circuit analysis. It requires a multimodal large model to analyze the problems, provide predicted answers, and compare them with the correct answers to generate accuracy rates. The specific generation format is as follows:

Example:

For Figure 3, the relevant questions are as follows:

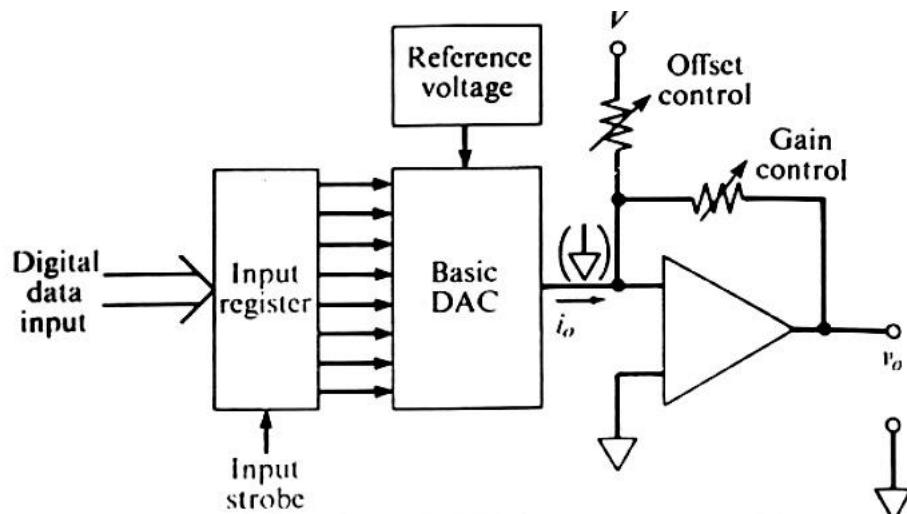


Figure 3

Multiple-choice question

1. Consider that the current output of the Basic DAC in the figure is input to the inverting terminal of the operational amplifier through an R_f adjustable feedback resistor (Gain control), and there is an equivalent capacitance C_{in} at the inverting input node. Which of the following statements correctly describes the relationship between the closed-loop bandwidth f_c and the gain control (feedback resistance value)?
 - A. Regardless of the value R_f taken, the closed-loop bandwidth remains unchanged because the open-loop bandwidth of Op Amp dominates the response.
 - B. As R_f increases, the closed-loop bandwidth decreases, roughly satisfying an

$$f_c \approx \frac{1}{2\pi R_f C_{in}} \text{ inverse relationship.}$$

- C. As R_f increases, the closed-loop bandwidth also increases because higher gain is beneficial for high-frequency response.
- D. As R_f increases, the closed-loop bandwidth first increases and then decreases, showing a non-monotonic change. This is caused by the coupling of gain control and Offset control.

Answer: B

If the model predicts the answer to be C, it is regarded as a matching error and the corresponding score cannot be obtained.

Based on the circuit shown in the figure, suppose the output current of the Basic DAC is I_o . In the end-to-end resistance of the Offset control of the bias potentiometer, the equivalent resistance connected between the reference voltage V_+ and the inverting input node of the operational amplifier is R_{off} , and the feedback resistance Gain control is R_f . Then, regarding the relationship between the output voltage v_o and I_o , V_+ , R_f and R_{off} , which of the following is correct?

A. $v_o = -I_o R_f + \frac{V_+}{R_{off}} R_f$

B. $v_o = -I_o R_f - \frac{V_+}{R_{off}} R_f$

C. $v_o = I_o R_f - \frac{V_+}{R_{off}} R_f$

D. $v_o = I_o R_f + \frac{V_+}{R_{off}} R_f$

Answer: B

If the model predicts the answer to be B, it is regarded as a correct match and the corresponding score can be obtained.

Fill-in-the-blank question

Consider that the current output of the Basic DAC in the figure is input to the inverting terminal of the operational amplifier through the adjustable feedback resistor R_f (Gain control), and there is an equivalent capacitance C_{in} at the inverting input node.

Please answer: With R_f _____, the closed-loop bandwidth decreases.

Answer: Increase

If the model's prediction result shows an increase, it is regarded as a successful match and the corresponding score can be obtained.

5. Scoring criteria

(1) Constraints

If the participating teams use VLM, the type of the inference large model they use should be: Qwen2.5-VL-3B. The number of Adapter parameters should be less than 1B, and the total calculation model parameters should be less than 4B. Participation using other large models will be regarded as invalid. Note: This requirement is only applicable to the testing process.

(2) Scoring method

Set several parameters for calculating the F1 score:

TP (True Positive): A positive example that is correctly predicted. That is, the situation where the true value of the data is a positive example and the predicted value is also a positive example;

TN (True Negative): A counterexample that is correctly predicted. That is, the situation where the true value of the data is a counterexample and the predicted value is also a counterexample;

FP (False Positive): A positive example that was wrongly predicted. That is, the situation where the true value of the data is a counterexample but is wrongly predicted as a positive example;

FN (False Negative): A counterexample that was wrongly predicted. That is, the situation where the true value of the data is a positive example, but it is wrongly predicted as a negative example.

According to the output part set by the competition question, each participating team will be scored in the form of a weighted average for this competition question. Suppose S1, S2, and S3 are the scores of the contestants in the output (1), (2), and (3) respectively. Here, the scores of S1, S2, and S3 are set at different scoring points, and each correct match of data will result in a corresponding score.

Task One (60%) :

Component extraction scoring criterion S1:

Match box: {[Component:,pos:], [...] The position information in} is calculated based on the target detection parameter IoU (intersection and union ratio). When $\text{IoU} \geq 0.5$, it is considered that the position matching is correct. If there are multiple boxes with $\text{IoU} \geq 0.5$, the component with the highest IOU is taken as the matching component, and the corresponding score can be obtained.

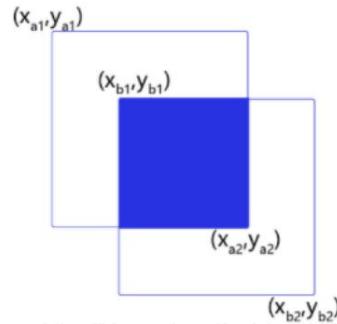


Figure 4 Calculation Method of IOU

The coordinates of the upper left corner of the intersection part are:

$$x_1 = \max(x_{a1}, x_{b1}), \quad y_1 = \max(y_{a1}, y_{b1})$$

The coordinates of the lower right corner of the intersecting part are:

$$x_2 = \min(x_{a2}, x_{b2}), \quad y_2 = \min(y_{a2}, y_{b2})$$

Calculation of intersection area:

$$\text{intersection} = \max(x_2 - x_1 + 1.0, 0) \cdot \max(y_2 - y_1 + 1.0, 0)$$

The areas of the two boxes:

$$S_A = (x_{a2} - x_{a1} + 1.0) \cdot (y_{a2} - y_{a1} + 1.0)$$

$$S_B = (x_{b2} - x_{b1} + 1.0) \cdot (y_{b2} - y_{b1} + 1.0)$$

Union area:

$$\text{union} = S_A + S_B - \text{intersection}$$

IoU calculation:

$$\text{IoU} = \frac{\text{intersection}}{\text{union}}$$

According to the parameters mentioned earlier, it can be obtained that:

TP: Number of correctly matched positions FP: Number of positions in the Label FN:

Number of positions generated by the model

Then the calculation formula is:

Accuracy

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall rate

$$\text{Recall} = \frac{TP}{TP + FN}$$

Location matching score

$$S_p = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}$$

After completing the calculation on SP, the model needs to generate a model name and match the component names using the method of **text editing distance** or **word vector calculation** for similarity. For components with obvious names in the figure, it is required to identify the names of the components and complete the correct matching. For components without obvious names, it is required to name the components according to their types and correctly match them to the corresponding components. The competition questions set a fixed threshold. If the similarity exceeds the threshold, it is regarded as a correct match.

The results generated by the model can be classified as:

TP: Number of correctly matched names
 FP: Number of names in the Label
 FN: Number of names generated by the model

Then the calculation formula is:

Accuracy

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall rate

$$\text{Recall} = \frac{TP}{TP + FN}$$

Name matching score

$$S_N = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}$$

Then the calculation formula for the score S_1 extracted by the component is as follows:

$$S_1 = \frac{(S_N + S_P)}{2}$$

Component input/output port number matching scoring criterion S2:

Match I_O: {[Component:,input:,output:,],[...]} In}, the number of ports for each component, when both the input port and the output port are correctly matched, can earn the corresponding score. (Note: When only one end of the input port and the output port matches correctly while the other end matches incorrectly, the corresponding score cannot be obtained.)

According to the parameters mentioned earlier, it can be obtained that:

TP: The number of components correctly matching the port
 FP: The number of components in the Label
 FN: The number of matches generated by the model.

Then the calculation formula is:

Accuracy

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall rate

$$\text{Recall} = \frac{TP}{TP + FN}$$

The calculation formula for the input/output port number matching score S_2 is as follows:

$$S_2 = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}$$

Connection relationship extraction scoring Criterion S3:

Match each Connection relationship pair, such as the connection relationship pair of E: Connection[input:["C","D"], output:["F","G"]]. For each correctly matched connection relationship pair, the corresponding Case score is obtained.

According to the parameters mentioned earlier, it can be obtained that:

TP: The number of correct matches FP: The number in the Label FN: The number of matches generated by the model.

Then the calculation formula is:

Accuracy:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall rate:

$$\text{Recall} = \frac{TP}{TP + FN}$$

The calculation formula for the matching score S3 of input and output port numbers is as follows:

$$S_3 = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}$$

After calculating the three-part scoring criteria for Task One, the overall scoring criteria for Task One will be calculated.

Task One Overall Scoring Criteria Score1:

$$Score_1 = S_1 \times 40 + S_2 \times 20 + S_3 \times 40$$

Task Two (40%) :

Q&A pair generation:

Calculate the accuracy rate of Q&A matching as the scoring criterion for this part.

Let T be the correct match, N be the incorrect match, and the accuracy rate be PQ. Then the scoring criterion formula for this part is:

Accuracy rate

$$P_Q = \frac{T}{T + N}$$

Score for Task Two:

$$\text{Score}_2 = P_Q \times 100$$

The model calculates the total Score scoring criterion Score:

In the process of calculating the total score, to prevent extreme scores from causing excessive differences in the rankings of each participating team, the competition question uses Min-Max normalization, mapping the scores within the interval [0,1]. The calculation formula is:

$$\text{Normalized}_x = \frac{x - \min(A)}{\max(A) - \min(A)}$$

Among them, x is the score obtained by the participating team in task x, for example, x1 is the score obtained by the participating team in Task One. $\min(A)$ represents the lowest score among all participating teams, and $\max(A)$ represents the highest score among all participating teams.

For the task scoring requirements of the total score, there are relative scores Normalized_{x1} and Normalized_{x2} , which respectively correspond to the relative scores obtained by the participating teams in Task One and Task Two.

Then the calculation formula for the total evaluation score is:

$$\text{Score} = \text{Nomalized}_{x_1} \times 60 + \text{Nomalized}_{x_2} \times 40$$

Model calculation time requirements

The calculation time of the model will be a bonus point for the competition question scoring. Regarding the calculation time, there are the following scoring methods:

The competition questions are calculated based on the average time AVG evaluated by Benchmark

$$AVG = \frac{\text{sum}(\text{runtime}(\text{Benchmark1}) + \text{runtime}(\text{Benchmark2}) + \text{runtime}(\text{Benchmark3}) + \dots)}{\text{count}(\text{Benchmark})}$$

The scoring experts analyzed the size of the AVG and gave the scores obtained by the participating teams in terms of time according to Table 1

Table 1 Calculation Time Scoring Table

AVG Team Ranking	Score
Top 10%	5 points
Top 30%	3 points
Top 50%	1 point
Other	0 point

For example, in Table 2, there are:

Table 2

Model calculation time /s	Team1	Team2	Team3	Team4	Team5
Benchmark1	40	50	20	80	120
Benchmark2	1780	6800	5000	3000	2300
Benchmark3	2000	3000	500	1200	5000
Benchmark4	3000	4000	2000	5000	3000
AVG	1705	3462.5	1880	2320	2605

The results of AVG can be obtained based on the calculation formula as follows:

$$AVG_1 = (40 + 1780 + 2000 + 3000) / 4 = 1705 \text{ s}$$

$$\text{AVG}_2 = (50 + 6800 + 3000 + 4000) / 4 = 3462.5 \text{ s}$$

$$\text{AVG}_3 = (20 + 5000 + 500 + 2000) / 4 = 1880 \text{ s}$$

$$\text{AVG}_4 = (80 + 3000 + 1200 + 5000 + 2320) / 4 = 2320 \text{ s}$$

$$\text{AVG}_5 = (120 + 2300 + 5000 + 3000) / 4 = 2605 \text{ s}$$

For Teams 1 to 5, the AVG scoring order is Team1 < Team 3 < Team 4 < Team 5 < Team 2.

Then, Team1 gets 5 points, Team 3 and 4 get 1 point, and Team 5 and 2 get no points.

Model Case Evaluation

For the evaluation Case of the model, the competition question is set as two parts: Public Case and Hidden Case. The participating teams can evaluate the Public Case to determine whether the performance of the model meets the requirements of the competition question.

After the participating teams submit their model codes, the organizing committee will use Hidden Case to evaluate the models and further analyze the performance of each participating team's models. The distribution ratio of Public cases and Hidden cases is shown in Table 3:

Table 3 Distribution Table of Model Evaluation Cases

Type	Public Case	Hidden Case
Proportion	60%	40%

6. Requirements for participants

Students are required to have a certain level of Python programming skills, as well as a basic understanding of logic circuits and computers. It is mainly targeted at students majoring in computer science, microelectronics and mathematics, but not limited to these majors.

7. References

Students can refer to the following references to review some existing studies and optimize the current models.

The following books cover knowledge of integrated circuits and multimodal large models:

- [1] Y. Huang et al., “LayoutLMv3: Unified Text and Image Masking for Document AI,” in Proc. ACM Multimedia, 2022, pp. 1–9.
- [2] J. Li et al., “Qwen-VL: A Frontier Large Vision-Language Model with Versatile Abilities,” arXiv:2308.12966, 2023.
- [3] Chen, T., et al. “Visual Instruction Tuning: Towards General-Purpose Multimodal LLMs.” NeurIPS 2023.
- [4] Li, J., et al. “Blip-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders.” ICML 2023.
- [5] Liu, S., et al. “LLaVA-OneVision: Easy Visual Task Transfer with Only One Vision-Language Model.” arXiv:2408.03326, 2024.
- [6] Deng, L., et al. “Multimodal Document Intelligence: Models, Benchmarks and Challenges.” ACM Computing Surveys, 2024.
- [7] LayoutLLM-3B: Lightweight Multimodal Transformer for Technical Diagram Understanding arXiv:2405.13214, 2024
- [8] L. Wang et al., “EDA in the Era of Large AI Models: Opportunities and Challenges,” IEEE Trans. CAD, early access, 2024.
- [9] X. Dong et al., “Towards Efficient Large Multimodal Model Serving: A System Perspective,” arXiv:2402.14818, 2024.

- [10] R. Kandur, “Multimodal Large Language Models: Architectures, Challenges and Future Directions,” *Int. J. Inf. Technol. Manag. Inf. Syst.*, vol. 16, no. 1, pp. 430–441, 2025.
- [11] S. Bai et al., “Qwen2.5-VL Technical Report,” arXiv preprint arXiv:2502.13923, Feb. 2025.

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