

“中国电子杯”第三届高校ICT产教融合创新大赛

赛题二

一、赛题名称

模拟电路 AI 智能化设计

二、命题单位

北京华大九天科技股份有限公司

三、所需知识背景

1. 专业背景不限。本赛题核心是基于华大九天泛模拟全流程平台及 PyAether 全定制设计平台生态系统，构建能理解自然语言、自主完成模拟电路拓扑模块识别与进行尺寸优化的模拟 AI 智能化设计。
2. 适合微电子、计算机、人工智能、电子信息、自动化、数学等相关专业的同学参加；鼓励跨学科组队等。
3. 参赛者团队需有 Python 语言，或者 C 语言，或者 C++ 语言、或者 Matlab 等的编程经验。掌握模拟电路设计、EDA 工具、Python / AI 基础，了解智能体框架，鼓励跨学科知识融合。

四、赛题背景

4.1 模拟电路设计自动化智能化的背景

现代模拟集成电路设计正向智能化、自动化演进，传统流程依赖工程师手动解读 Spec、选择拓扑、迭代尺寸，效率低、周期长、对经验依赖度高。面向自然语言性能规格自动完成拓扑模块识别与尺寸优化，是 EDA 与模拟设计融合的核心方向，可大幅降低设计门槛、提升设计效率与方案鲁棒性。全差分运放是模拟电路的核心单元，广泛应用于数模转换器、滤波器、传感器和信号处理等场景。

传统设计需人工匹配拓扑与尺寸，本赛题通过智能体实现自然语言 Spec→拓扑模块识

别与参数约减→尺寸自动优化的端到端自动化，探索人机协同的模拟电路智能化设计新流程，推动国产 EDA 生态的智能化发展。

五、赛题描述

本赛题构建模拟电路智能设计体，实现三大核心能力：自然语言 Spec 解析、拓扑模块识别与参数约减、多目标尺寸优化，在给定 PyAether 自动化流程与 PDK 约束下，输出满足 Spec 的最优拓扑与尺寸方案，推动模拟设计智能化发展，具体模块说明如下。

序号	模块	说明
1	自然语言 Spec 解析（智能体）	输入：自然语言描述的性能约束（例如“设计全差分运放，相位裕度$\geq 60^\circ$，工作电流 3mA，增益尽可能大，单位增益带宽尽可能大，面积尽可能小”） 输出：结构化性能指标，包括硬约束项、优化目标、指标单位及目标优先级等信息。完成自然语言需求向结构化 Spec 的转换。整体为 Json 文件格式。
2	拓扑模块识别与参数约减（智能体）	输入：电路网表 要求：识别电路内部电流镜、输入对管、有源负载、匹配器件、Dummy 器件等核心功能模块与匹配结构，剔除冗余优化变量，完成参数约减，有效缩减优化变量数量，提升后续尺寸优化效率。 交付物：拓扑模块识别结果，参数约减之后的变量个数与约减之后的电路网表。
3	基于约束的多目标尺寸优化（九天服务器）	优化变量：经参数约减后的晶体管 W、L，电阻电容等器件 参数约束条件：PVT 条件下满足相位裕度 PM、增益裕度 GM、工作电流 I_OPA 等指标 要求优化目标：开环 DC 增益越大越好（不得低于 40dB）、最大化单位增益带宽 UGB，最小化芯片面积 Area

		优化算法推荐（仅供参考）： 差分进化、遗传算法、模拟退火、贝叶斯优化、机器学习 AI 算法或自研算法 输出结果： 最优器件尺寸参数；最优拓扑 + 尺寸反标电路图；全 PVT 仿真验证报告；完整呈现从自然语言 Spec 到电路实现全链路
4	PyAether 辅助自动化流程脚本（九天服务器）	参赛者可使用赛题提供的基础 PyAether 流程脚本，搭建全自动化设计链路： 1. 电路图器件尺寸转为待优化变量的参数化脚本 2. 读取电路器件连接关系脚本 3. 调用 MDE 完成 PVT 仿真、提取性能指标的仿真引擎脚本 4. 将最优尺寸回写至电路图的反标脚本 5. 高集成自动优化接口脚本，输入尺寸文件即可自动调用仿真输出性能结果

更详细赛题描述，见本 word 最后的“附件：参考资料”。

六、评价指标与阶段作品要求

6.1 通用说明

本题分为校内初赛、决赛小组赛两个阶段。

本次竞赛给定样本分为给定已知运放 Public 电路（1 个）和内部 Hidden 测试运放电路（1 个），已知 Public 运放电路向参赛者公开，测试验证所开发算法。内部 Hidden 测试运放电路不向参赛者公开，用于检验参赛者所开发算法的通用性。公开 Public 电路不参与评分，内部 Hidden 测试运放参与评分。单个运放电路总体代码的运行时间上限为 3 小时。

6.2 通用提交报告要求

参赛者需要通过使用 Python/C++ 等编程语言，结合 PyAether 接口来实现本题。需提交智能体全流程源码、完整设计文档及对应输出文件，所有内容需完整覆盖全设计

链路。

(1) **技术报告：**提交 Doc 版本及 PDF 版本各 1 份（≤80 页），建议封面或目录后的 1-2 页设置为核心内容预览页，展示作品核心亮点。报告需要包含，在智能体方面系统提示词（System Prompt），配置文件，推理链日志（Chain-of-Thought），记录每一步决策依据：为何选择该拓扑、为何约减掉某参数、为何保留某变量；最终 Spec 解析结果，拓扑模块识别与参数约减方案等；在尺寸优化方面，优化算法设计架构、算法原理、优化结果对比（含电路图截图、仿真曲线）、全套测试报告等，完整覆盖 Spec 解析、模块识别、参数约减、仿真流程、优化算法全链路，附带所有输出成果文件说明。

主要包括：

序号	输出内容	格式要求
1	Spec 解析	从自然语言到 Spec 的结构化语言，输出的 Json 文件
2	拓扑模块识别与参数约减成果	模块识别依据及结果，参数约减之后的变量个数、变量统计列表，与约减之后的网表
3	器件识别之后的变量个数和带变量的电路图	CSV 文件（如 M1_w=36u, M1_L=2u）和带变量的 schematic
4	带最优的器件尺寸数值的反标后电路图	.schematic_var_best（含最优尺寸）
5	仿真验证报告	最优尺寸下的仿真结果和波形（含 TYP 和 PVT 结果），包括全 PVT 仿真结果、波形、性能达标情况；需覆盖工艺 TT/FF/SS、高温（125℃）、低温（-40℃）、标称电压±10%波动的全场景仿真结果
6	代码质量与算法性能	包括优化运行次数、运行时间、占用内存等

(2) **源代码：**提交压缩包并上传至华大九天服务器平台，包含智能体流程 Python 源

码，覆盖 Spec 解析、拓扑模块识别与参数约减等功能；华大九天平台上源代码（Python/C++），覆盖参数化电路图标注、基于约束的多目标尺寸优化等功能。

(3) 汇报 PPT：电子文档（≤20 页），内容包含项目背景、核心设计方案、主要工作内容、测试结果与数据分析。

(4) 演示视频：视频文档（≤5 分钟），内容包含参赛、核心设计方案、主要工作内容、测试结果与数据分析。MP4 格式，时长控制在 5 分钟以内，完整展示设计全流程与成果效果。

6.3 详细客观性能打分标准

(1) 【20 分】Spec 解析与模块识别参数约减

自然语言 Spec 正确解析为 Json 结构化指标（10 分）。Json 文件需要包括硬约束项、优化目标、指标单位及目标优先级等信息，每缺少一个扣 1 分，直至本项扣为 0 分。

拓扑模块识别与参数约减（10 分）。需要具有拓扑模块识别结果，参数约减之后的变量个数与约减之后的网表。每错一个约束，则扣 1 分，直至本项扣为 0 分。

(2) 【30 分】基于给定电路的约束条件的满足

通过给定的电路图和 MDE 环境，以及 PyAether 源代码，满足对相位裕度 PM、增益裕度 GM 和直流增益 DC Gain，以及工作电流 I_OPA 的最基本的约束条件。如有约束不满足，例如相位裕度不满足，则本项直接扣除 10 分，即该项分数为 0。如果满足，则直接给满分。

约束	分数	约束条件
相位裕度 PM≥50°	10 分	所有 PVT 均需满足
增益裕度 GM≤-10dB	10 分	所有 PVT 均需满足
工作电流 I_OPA≤3mA	10 分	所有 PVT 均需满足

(3) 【40 分】多目标尺寸优化算法之优化目标评比

综合单位增益带宽 UGB，和初步面积加和等指标，打分器将根据以下几个指标进行评

判最终结果的优劣：

目标	分数	评比条件
直流增益 DC Gain 越大越好	10 分	所有 PVT 条件下的最坏情况。一定需要大于等于 40dB，小于 40dB 则本项记为 0 分
单位增益带宽 UGB 越大越好	15 分	所有 PVT 条件下的最坏情况
面积最小化	15 分	注：面积参数，初步计算为所有晶体管和电阻电容面积的加和，其中： ● 每个晶体管的面积为 $f_w \cdot L \cdot m \cdot 1.5$ ● 电容面积为： $W \cdot L \cdot 1$ ● 电阻面积为： $W \cdot L \cdot 2$

每项目标按照排名得分：排名第一名本项为满分；以第一名排名为基准，其他按照比例进行计算得分。

直流增益 DC Gain 小于 40dB 的，则本项 Gain 记为 0 分。大于等于 40dB 的，以 DC Gain 最大的队伍记为 DC Gain0，其他团队的 DC Gain 指标记录里为 DC Gain1，则 DC Gain 得分为 $(DC\ Gain1/DC\ Gain0) \cdot 10$ 。

在满足 DC Gain 大于 40dB 的前提下，以 UGB 最大的队伍的指标记为 UGB0，其他团队的 UGB 指标记录为 UGB1，则 UGB 得分为 $UGB1/UGB0 \cdot 15$ 。

在满足 DC Gain 大于 40dB 的前提下，以面积最小的队伍的作为满分，最小面积记为 A0，其他团队的面积，例如为 A1，面积得分为 $A0/A1 \cdot 15$ 。

(4) 【10 分】 总体运行总时间，0~10 分。

基础得分：以满足 3 项约束条件的队伍中的最短运行时间，记为 T0，其他团队的运行时间记为 T1，则该团队的运行时间的基础得分为 $T0/T1 \cdot 10$ ，最高基础得分为 10 分。

每当有一项约束条件不满足，则本项扣 5 分。

6.4 各阶段评分标准

6.4.1 校内初赛阶段——技术报告+客观性能+基础知识问卷

大项	内容	分值	评分要求
----	----	----	------

大项	内容	分值	评分要求
通用提交报告要求	技术报告，源代码，汇报 PPT 和演示视频等	20 分	详细见“6.2 通用提交报告要求”，重点考察各项提交报告的完整性。报告结构清晰，逻辑严谨，图表规范美观，准确阐述设计思路、技术难点与解决方案，数据标注准确无误等。
客观性能	整体各项客观性能（在多约束下多目标优化）	70 分	详细见“6.3 详细客观性能打分标准”，重点考察客观智能体能力与尺寸优化结果。每项根据客观评价标准打分。
PyAether 基础知识问卷	进行基础知识问卷作答	10 分	作为本参赛基础技术掌握情况，提供基于 PyAether 的华大九天基础知识问卷二维码，参赛者进行客观作答，均需作答。以参赛者团队所有成员中，获得的最高分数，作为本团队分数。 例如，参赛团队包含 A,B,C 三人，需要 A,B,C 三人全部作答。取三者最高分，例如 89 分。则本项成绩为： $89/100 \times 10 = 8.9$ 分。

6.4.2 决赛阶段——技术报告+客观性能 + 创新设计 + 现场答辩

大项	内容	分值	评分要求
通用提交报告要求	技术报告，源代码，汇报 PPT 和演示视频等	12 分	详细见“6.2 通用提交报告要求”，重点考察各项提交报告的完整性。报告结构清晰，逻辑严谨，图表规范美观，准确阐述设计思路、技术难点与解决方案，数据标注准确无误等。
客观性能	整体各项客观性能（在多约束下多目标优化）	68 分	详细见“6.3 详细客观性能打分标准”，重点考察客观智能体能力与尺寸优化结果。每项根据客观评价标准打分。
答辩表现	汇报与问答	10 分	表达清晰流畅，能准确回答评委提问，展现扎实的专业基础
整体创新性	综合评定参赛作品能力，方案优化与拓展等	10 分	方案创新性、系统性、实用性及团队协作表现等

八、参赛平台

- 1. 第 1,2 智能体部分，本地部署，输出结果；
- 2. 智能体指定大语言模型：Qwen 3.8-Max，使用 API Key 调用方式；
- 3. 第 3,4 部分，远程登录华大九天 VNC，Linux Centos 服务器环境；
- 平台预配置 Pycharm 开发环境；
- 参赛者需通过 Python 等编程语言完成赛题全流程开发；
- 配套工具：ALPS 仿真器、Aether 全定制设计平台、PyAether 生态系统。

附件：参考资料

1) 自然语言输入示例

输入示例：“请设计一个带有共模反馈的全差分运放，要求：DC 增益不低于 95dB，单位增益带宽至少 60MHz，相位裕度大于 55 度，所有 PVT 下工作电流不超过 3mA，并尽量减小面积。”

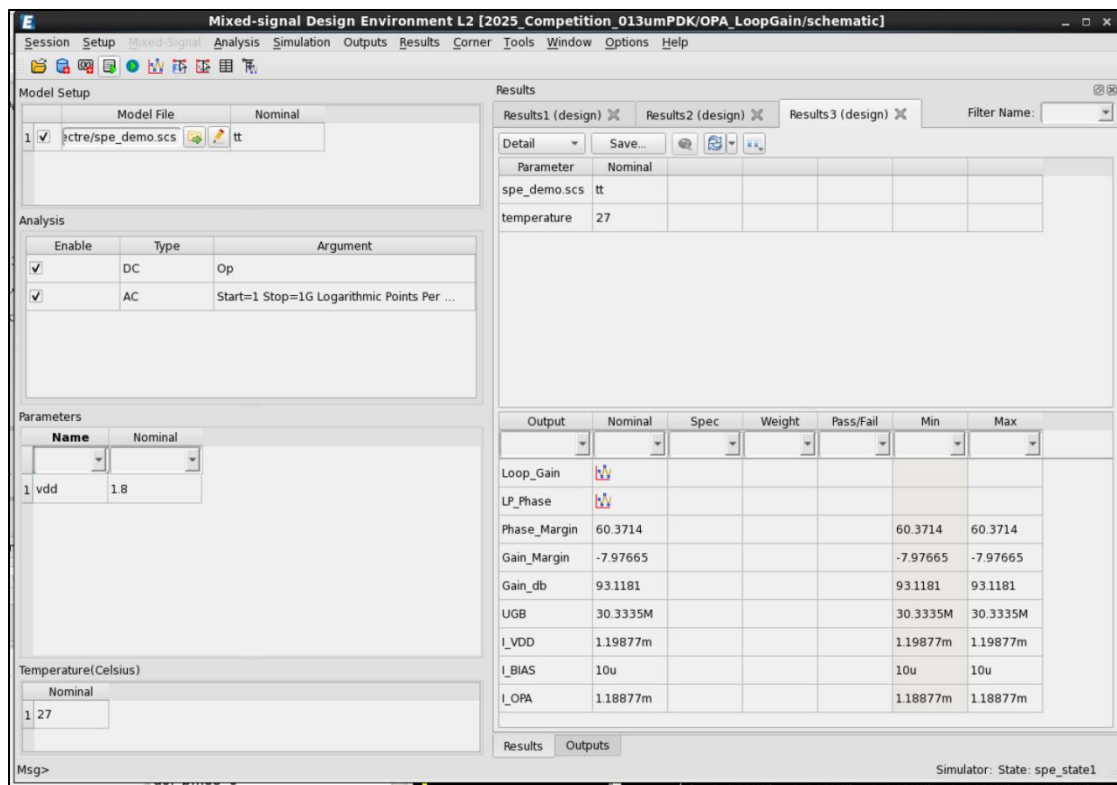
解析输出：JSON 格式文件。

2) 电路拓扑与器件参数约束

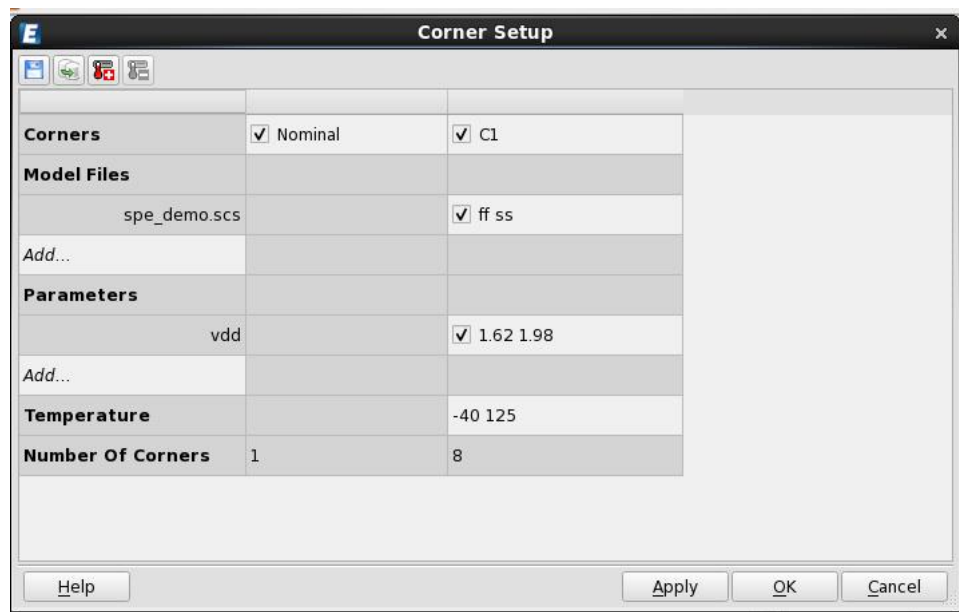
赛题提供运放拓扑结构，通过 PyAether 参数化脚本，可将电路器件尺寸转化为可优化变量，所有变量需严格遵循工艺取值约束。

器件类型	参数	取值范围
晶体管	宽度	0.13 μm – 20 μm
	长度	0.13 μm – 10 μm
	Multiplier(m)	1,2,...20(整数)
电阻	Seg (Segment)	1,2,...20(整数)
电容	L 值	0.13 μm – 20 μm

包括单次仿真的设计环境和输出结果等，输出结果包括增益 Gain 和相位 Phase，单位增益带宽 UGB 和相位裕度 PM、增益裕度 GM，工作电流 I_OPA 等。



以及需要满足的 PVT 条件的设置，如下图所示：



4) 提供基于 PyAether 的“将电路图参数转变成变量”的源代码

通过 PyAether 和已知的电路图，运行该脚本，可以自动生成参数化电路图，即将已知电路图中的晶体管、电阻、电容的尺寸转化为变量。核心代码如下：

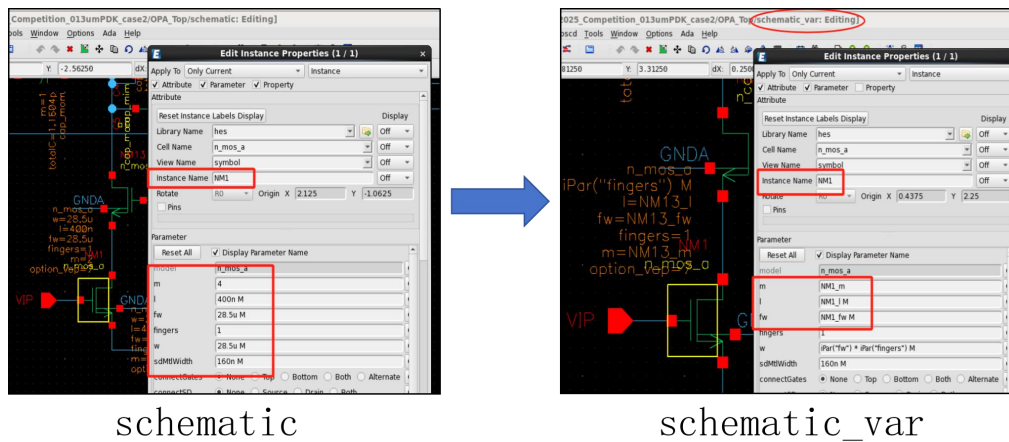
```

run_sim_change_sch_var.py (/home/works/cxuser27) - GVIM
File Edit Tools Syntax Buffers Window Help

14
15 def set_para(lib, cell, view):
16     cv_source = pyScript.dbOpenCV(lib, cell, view)
17     if(cv_source is None):
18         print("open failed")
19     ns = pyScript.emyCdbANS()
20     libS = pyScript.emyScalarName(ns, str(lib))
21     Cells = pyScript.emyScalarName(ns, str(cell))
22     Views = pyScript.emyScalarName(ns, str(view)+"_var")
23     cv_source.saveAs(libS, Cells, Views)
24
25
26 cv = pyScript.dbOpenCV(lib, cell, view+"_var")
27 insts = cv.instances
28 for inst in insts:
29     print("====start ", inst.name, "====")
30     if (inst.name[0] == "R" or inst.name[0] == 'r'):
31         pyScript.cdfSetParam(inst, 'segments', inst.name + '_seg')
32     elif (inst.name[0] == "C" or inst.name[0] == 'c'):
33         pyScript.cdfSetParam(inst, 'm', inst.name + '_m')
34     elif (str(inst.name).startswith("PIN") or str(inst.name).startswith("pin")):
35         print("====this is a pin ", inst.name, "====")
36     else:
37         m_return = pyScript.cdfSetParam(inst, 'm', inst.name + '_m')
38         l_return = pyScript.cdfSetParam(inst, 'l', inst.name + '_l')
39         fw_return = pyScript.cdfSetParam(inst, 'fw', inst.name + '_fw')
40         print("m_return: ", m_return)
41         print("l_return: ", l_return)
42         print("fw_return: ", fw_return)
43
44     print("====finish ", inst.name, "====")
45 # dbcheckandsave()
46 pyScript.dbCheckAndSaveDesign(cv)
47 # cv.save()

```

运行本 PyAether 脚本之后，可以生成带有各种变量的 schematic_var 电路图。这些变量，主要通过 instance 名字加_cdf 参数形成，例如 NM1_m 等。



5) 提供基于 PyAether 的“调用 MDE 进行仿真”的源代码

通过 PyAether 和已知的 MDE，运行该脚本，可以自动调用相应的仿真，并且得到仿真结果。运行效果如下：

```

-----
Simulation run started.
Results4 Sim Done! #maxJob: 1 #total sim: 13 #finished sim: 1
Results4 Sim Done! #maxJob: 1 #total sim: 13 #finished sim: 2
Results4 Sim Done! #maxJob: 1 #total sim: 13 #finished sim: 3
Results4 Sim Done! #maxJob: 1 #total sim: 13 #finished sim: 4
Results4 Sim Done! #maxJob: 1 #total sim: 13 #finished sim: 5
Results4 Sim Done! #maxJob: 1 #total sim: 13 #finished sim: 6
Results4 Sim Done! #maxJob: 1 #total sim: 13 #finished sim: 7
Results4 Sim Done! #maxJob: 1 #total sim: 13 #finished sim: 8
Results4 Sim Done! #maxJob: 1 #total sim: 13 #finished sim: 9
Results4 Sim Done! #maxJob: 1 #total sim: 13 #finished sim: 10
Results4 Sim Done! #maxJob: 1 #total sim: 13 #finished sim: 11
Results4 Sim Done! #maxJob: 1 #total sim: 13 #finished sim: 12
Results4 Sim Done! #maxJob: 1 #total sim: 13 #finished sim: 13
-----Result is valid-----
True
-----
-----Result name-----
Results4
-----

```

6) 提供基于 PyAether 的“将最优电路尺寸数据反标到电路图”的源代码

通过 PyAether 和本次考核的优化算法得到的最佳尺寸参数 csv 文件，运行该脚本，可以自动生成最佳尺寸的电路图。核心代码如下：

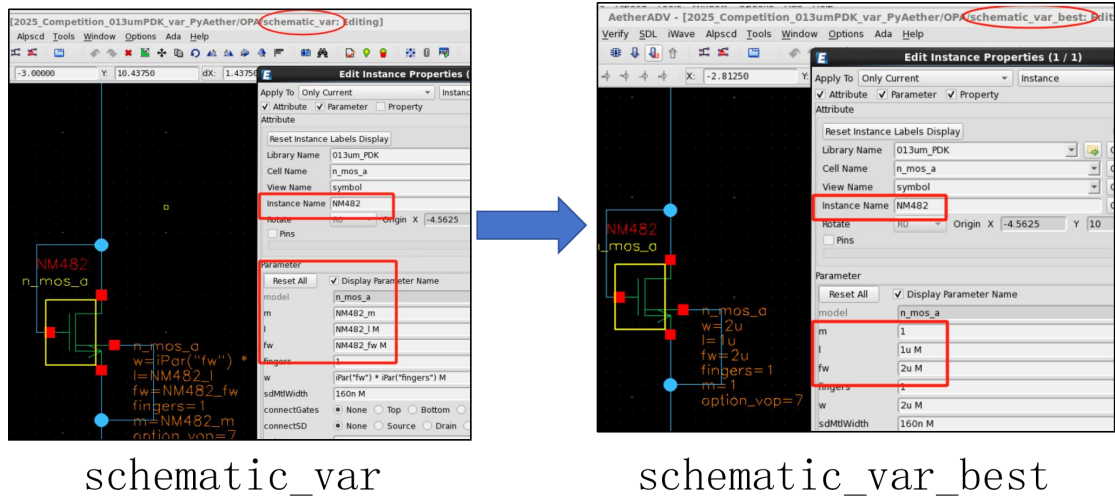
```

run_sim_get_best_size.py (/home/works/cxuser27) - GVIM1
File Edit Tools Syntax Buffers Window Help

13
14 def get_para(lib, cell, view):
15
16     cv_source = pyScript.dbOpenCV(lib, cell, view)
17     if(cv_source is None):
18         print("open failed")
19
20     ns = pyScript.emyCdbANS()
21     libS = pyScript.emyScalarName(ns, str(lib))
22     Cells = pyScript.emyScalarName(ns, str(cell))
23     ViewS = pyScript.emyScalarName(ns, str(view)+"_best")
24     cv_source.saveAs(libS, Cells, ViewS)
25
26
27     cv = pyScript.dbOpenCV(lib, cell, view+"_best")
28
29     with open('get_pa_result.csv', 'r') as result_file:
30         lines = result_file.readlines()
31         for line in lines:
32             print("proc ", line)
33             fields = line.strip().replace('""', '').split(',')
34             print("fields[0]: ", fields[0])
35             if (fields[0] == "parameter"):
36                 print("fields[1]: ", fields[1])
37                 if (fields[1] == "vdd"):
38                     continue
39                 index = fields[1].rfind(" ")
40                 inst_name = fields[1][:index]
41                 param_name = fields[1][(index+1):]
42                 print("inst name: ", inst_name)
43                 print("param name: ", param_name)
44                 instId = pyScript.dbFindInst(cv=cv, name=inst_name)
45                 print("inst ID: ", instId)
46                 pyScript.cdfSetParam(instId, param_name, fields[2])
47 #     cv.save()
48     pyScript.dbCheckAndSaveDesign(cv)

```

最终生成含有最佳尺寸的 schematic_var_best 电路图。

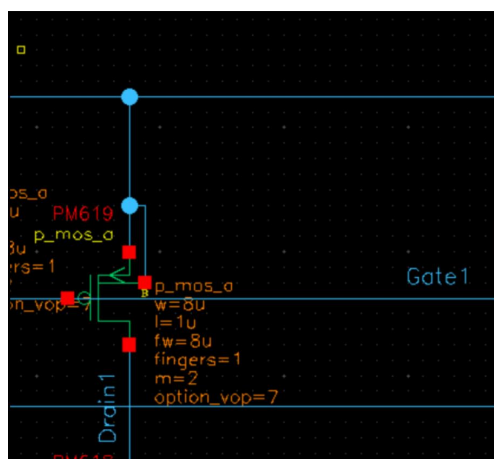


7) 提供基于 PyAether 的“读取器件的连接关系”的源代码

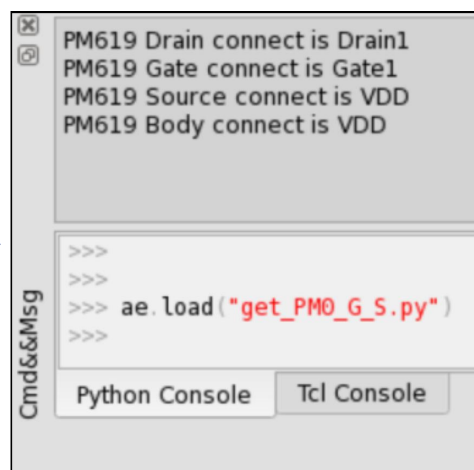
通过给定的 PyAether 代码，可以读取电路图中器件的连接关系，可以自动打印出器件的 Gate, Source 等所连接的 Net 的名字。核心代码如下：

```
get_PM0_G_S.py = (/home/PyAether_Packages/Desktop/2025_EDA_Competition) - GVIM5
File Edit Tools Syntax Buffers Window Help
2
3 #open design
4 cv = ae.dbOpenCV(lib="2025_Competition_013umPDK_var_PyAether", cell="OPA", view="schematic")
5
6 I_PM0 = ae.dbFindInst(cv, "PM619")
7
8 for p in I_PM0.instTerms:
9     if p.name=="G":
10         print("PM619 Gate connect is", p.net.name)
11     elif p.name=="S":
12         print("PM619 Source connect is", p.net.name)
13     elif p.name=="D":
14         print("PM619 Drain connect is", p.net.name)
15     else:
16         print("PM619 Body connect is", p.net.name)
```

运行效果如下：



schematic



读取的连接关系

8) 提供基于 PyAether 的整体的、参赛者只需要输入尺寸文件的自动优化接口的源代码

最终，参赛者仅需要调用以下命令 `run_opt_command.sh`，即可在相应尺寸下，调用仿真，得到仿真结果。

`run_opt_command.sh` 的命令如下：

`pythone optimization.py`

`--ae_lib input_Competition_lib_name`

`--ae_cell input_OPA`

`--ae_view input_schematic`

`--mde_cell input_OPA_LoopGain`

`--mde_view input_spe_state1_corner`

`--output_path ./results_case1`

`--output_file output.log`

`--param_file extractedcdfVal_0.txt`

其中：

蓝色部分为赛题输入部分，已经设置好：

--ae_lib: 输入信息, 待优化的模拟电路的 Lib 名字, 已经设置好

--ae_cell: 输入信息, 待优化的模拟电路的 Cell 名字, 已经设置好

--ae_view: 输入信息, 待优化的模拟电路的 View 名字, 已经设置好

--mde_cell: 输入信息, 待优化电路的 TestBench, 已经设置好

--mde_view: 输入信息, 待优化电路的 TestBench 所对应的 MDE L2,已经设置好

绿色部分为仿真结果的输出结果, 可以自行定义输出结果的路径:

--output_path: 仿真结果的输出路径

--output_file: 所有仿真结果 (增益, 裕度, UGB 等) 的输出总结

红色部分为每次仿真所需输入的电路尺寸参数。

--param_file: 输入文件, 这里展示的是输入的初始电路尺寸参数文件:

extractcdfVal_0.txt; 参赛者需要根据自己所使用的优化算法, 得到新的电路尺寸文件, 来代替这个.TXT 文件, 从而得到新的仿真结果。

注意: 本赛题需要在给定的初始解条件下, 进行优化。

其运行如下所示:

```
1104 ./run_opt_command_case3.sh
1105 history
empyrean@jingying01 2025_EDA_Competition]$ ./run_opt_command.sh
```

可以得到的仿真结果的文件如下所示:

```
[empyrean@jingying01 2025_EDA_Competition]$ cd results_case3
[empyrean@jingying01 results_case3]$ ls
emy2netlist.log  Results3_detail.csv  Results3_transpose.csv  Results4_detail.csv  Results4_transpose.csv
output_case3.log Results3_summary.csv  Results3_yield.csv      Results4_summary.csv  Results4_yield.csv
```

可以查看仿真的总结结果:

```
Results4_summary.csv (/home/PyAether_Packages/Desktop/2025_EDA_Comp)
File Edit Tools Syntax Buffers Window Help
"Output", "Min", "Max", "Mean", "Median", "Std Dev", "Spec", "Pass/Fail", "Pass Ratio"
"Phase_Margin", "6.32965e+01", "8.14564e+01", "7.52414e+01", "7.54681e+01", "4.96858", "", "", ""
"Gain_Margin", "-8.25132", "-4.68257e-01", "-3.61279", "-2.63454", "2.40459", "", "", ""
"Gain_db", "6.06458e+01", "9.1732e+01", "7.84067e+01", "8.06458e+01", "1.0033e+01", "", "", ""
"UGB", "2.748e+07", "1.36736e+08", "7.6418e+07", "8.34911e+07", "3.05084e+07", "", "", ""
"I_OPA", "1.44366e-04", "4.59344e-04", "2.85842e-04", "2.7091e-04", "9.32624e-05", "", "", ""
```

可以查看每个仿真的具体结果：

```
Results4_detail.csv (/home/PyAether_Packages/Desktop/2025_EDA_Competition/results_case3) - GVIM3
File Edit Tools Syntax Buffers Window Help
"Parameter", "Nominal", "C1_0", "C1_1", "C1_2", "C1_3", "C1_4", "C1_5", "C1_6", "C1_7", "C1_8", "C1_9", "C1_10", "C1_11"
"spe_demo.scs", "tt", "ff", "ff", "ff", "ff", "ss", "ss", "ss", "ss", "ss", "ss"
"vdd", "1.8", "1.62", "1.8", "1.8", "1.98", "1.98", "1.62", "1.62", "1.8", "1.8", "1.98", "1.98"
"temperature", "2.7e+01", "1.25e+02", "1.25e+02", "1.25e+02", "1.25e+02", "1.25e+02", "1.25e+02", "1.25e+02", "1.25e+02", "1.25e+02", "1.25e+02"
"Output", "Nominal", "Spec", "Weight", "Pass/Fail", "Min", "Max", "C1_0", "C1_1", "C1_2", "C1_3", "C1_4", "C1_5", "C1_6", "C1_7", "C1_8", "C1_9", "C1_10", "C1_11"
"Loop Gain", "7.56757e+01", "6.32965e+01", "8.14564e+01", "6.32965e+01", "6.81177e+01", "7.39604e+01", "7.54681e+01", "8.06272e+01", "7.82741e+01", "7.3091e+01", "7.49137e+01"
"LP_Phase", "7.41177e+01", "7.79372e+01", "8.14564e+01", "8.12019e+01"
"Phase_Margin", "7.56757e+01", "6.32965e+01", "8.14564e+01", "6.32965e+01", "6.81177e+01", "7.39604e+01", "7.54681e+01", "8.06272e+01", "7.82741e+01", "7.3091e+01", "7.49137e+01"
"Gain_Margin", "-2.55077", "-8.25132", "-4.68257e-01", "-4.68257e-01", "-2.06157", "-2.63454", "-4.71907", "-8.25132", "-7.62014", "-1.24886", "-2.04139", "-1.27962", "-2.89042", "-5.41757", "-5.78275"
"Gain_db", "8.25921e+01", "6.06458e+01", "9.1732e+01", "9.09471e+01", "8.09052e+01", "7.95258e+01", "6.98823e+01", "6.34838e+01", "6.06458e+01", "9.1732e+01", "8.68052e+01", "9.02161e+01", "8.06458e+01", "7.34122e+01", "6.85036e+01"
"UGB", "8.79772e+07", "2.748e+07", "1.36736e+08", "1.12218e+08", "8.34911e+07", "6.78368e+07", "2.748e+07", "3.75415e+07", "9.2506e+07", "8.72317e+07", "9.69699e+07", "7.72054e+07", "4.2839e+07", "4.34008e+07"
"I_OPA", "2.7091e-04", "1.44366e-04", "4.59344e-04", "1.91186e-04", "3.52628e-04", "2.28413e-04", "4.09192e-04", "2.65485e-04", "4.59344e-04", "1.44366e-04", "2.8838e-04", "1.77496e-04", "3.38829e-04", "2.10911e-04", "3.86799e-04"
```

9) 提供工艺约束：

提供相关工艺 PDK 库。