2025 中国研究生创“芯”大赛·EDA 精英挑战赛

一、赛题名称

模拟电路自动尺寸优化设计--基于 PyAether 生态共塑 EDA 未来

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。本赛题的核心是能够通过基于全定制设计平台生态系统 PyAether，进行模拟电路尺寸的自动化优化设计。

2. 适合计算机、微电子、电子信息、自动化、数学等专业的同学参加。

3. 参赛者有过 Python 语言，或者 C 语言，或者 C++语言、或者 Matlab 等的编程经验。

四、赛题背景

4.1 模拟电路设计自动化的背景

在现代集成电路设计中，模拟电路设计是一个复杂且耗时的过程，尤其是对于高性能的模拟电路，如全差分高增益放大器。传统的人工设计方法依赖于工程师的经验和迭代优化，这不仅效率低下，而且容易出错。随着集成电路设计复杂度的增加，手动优化变得越来越困难。因此进行模拟电路自动尺寸优化，对于提高设计效率、缩短研发周期具有重要意义。

4.2 全差分高增益放大器的介绍

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，其主要特点是高增益、高共模抑制比等。其关键指标包括：开环直

流增益（Open-Loop DC Gain, eg, ≥100dB）以抑制噪声与失真；相位裕度（PM, eg, ≥50°）和增益裕度（GM, eg ≤-6dB）确保系统稳定性；电源抑制比（PSRR @DC≥112dB, PSRR @1MHz≥80dB）以抵抗电源波动，芯片功耗（eg, 工作电流 Isum ≤1mA，关断电流 Istandby_current≤100nA 等）以满足低功耗要求。全差分运放通常通过多级增益与米勒补偿等方式实现高增益与稳定性，但手工优化需兼顾晶体管宽长比、偏置电流、补偿网络等参数，存在着反复的多次迭代尝试

4.3 自动尺寸优化的重要性和必要性

自动尺寸优化是模拟电路设计中的关键步骤，它能够显著提高设计效率，减少设计周期，并提高电路性能的一致性。通过自动尺寸优化算法，可以在短时间内探索大量的设计变量，找到最优的尺寸组合，从而满足设计指标，如增益、带宽和电源噪声抑制比等。

- **效率提升**：通过算法替代人工试错，缩短设计周期；
- **性能优化**：基于多目标优化（如增益、带宽、稳定性,面积等）搜索全局最优解；
- **PVT 适配**：针对不同 PVT（工艺、电压、温度）自动调整参数，确保设计可靠性。

五、赛题描述

本赛题旨在考核模拟电路尺寸自动优化算法。通过给定的电路图和仿真设置，以及提供的 PyAether 相关源代码辅助流程自动化前提下，参赛者通过设计优化算法，实现模拟电路的尺寸自动优化，需要在满足约束条件的前提下，实现运放电路的性能最优；从而推动模拟电路设计自动化的发展，提高设计效率、缩短研发周期。

5.1 任务：基于约束条件的多目标尺寸优化算法开发

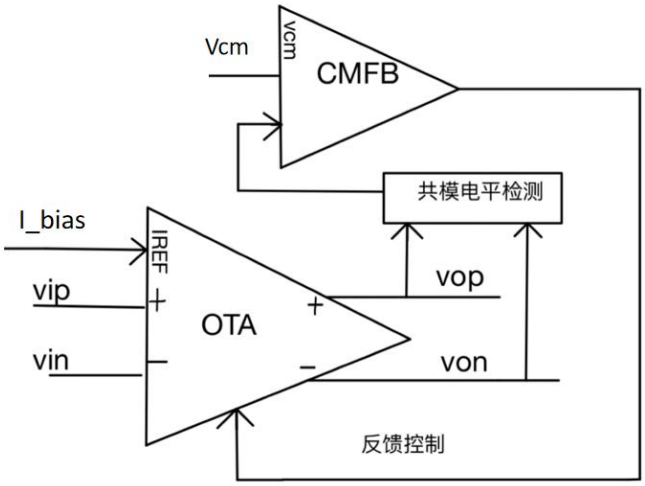
任务：进行器件识别，将设计变量个数进行缩减；设计优化算法，得到最优尺寸，以满足相应约束条件和最佳优化目标。

- **优化变量**：晶体管 W, L、电阻电容值等
- **约束条件**：相位裕度 PM，增益裕度 GM，开环 DC 增益 Loop Gain，和工作电流 I_OPA 需要在 PVT 条件下，都满足相应的要求。具体见评分环节第八章。
- **优化目标**：单位增益带宽 UGB，和初步面积（Area）评估。单位增益带宽越大越好，面积越小越好。具体见评分环节第八章。
- **器件识别与变量个数优化**：基于 PyAether，识别模拟 IC 中的匹配结构（例如电流镜、输入对管、有源负载、电阻匹配器件、电容匹配阵列、Dummy 器件等），从而显著减少优化变量数量，提升优化效率；需要得到器件识别之后的变量个数。
- **算法类型**：包括但是不局限于差分进化、遗传算法、模拟退火、贝叶斯优化、AI（人工智能算法，机器学习）等，也可以自研相关算法，得到最终的晶体管尺寸、电阻和电容等数值。
- **输出结果**：缩减之后的变量个数，以及基于优化算法，得到最佳尺寸，根据给定的 PyAether 相关脚本例子，输出最佳电路图，同时给出来最佳尺寸下的 PVT 仿真结果等。

5.2 将为参赛队伍提供赛题 Case：包括电路设计和仿真设置等

将为参赛队伍提供赛题 Case，包括多个全差分运放模拟电路图，这些电路图中包含核心运放（例如折叠式运放，和套筒式运放等，以及共模反馈电路等部分。以及包含激励在内的电路仿真测试电路图 Test Bench，以及模拟混合仿真环境 MDE，其主要

包括单次仿真的设计环境和输出结果等，（增益和相位，单位增益带宽和相位裕度、增益裕度，工作电流等），以及需要满足的 PVT 条件的设置等。



5.3 将为参赛队伍提供基于 PyAether 的辅助自动化流程相关脚本

将为参赛队伍提供基于 PyAether 的辅助自动化流程相关脚本。主要包括：

- 将电路图中器件尺寸转换为待优化变量的参数化 PyAether 脚本
- 读取器件的连接关系的 PyAether 脚本
- 能够调用 MDE 进行 PVT 仿真、输出性能指标的仿真引擎调用 PyAether 脚本
- 最终将最优尺寸写入电路图中的反标 PyAether 脚本
- 集成度更高的、参赛者只需要输入尺寸文件、即可调用仿真得到本尺寸仿真结果的自动优化接口的 PyAether 脚本

通过这些给定的将电路图中器件尺寸转换为待优化变量的参数化 PyAether 脚本，可以把原始电路中的器件转变成变量，例如晶体管宽长比（W/L）、电阻值和电容值等设为变量（如 M1_W=16u、Rc=10k）；所以优化算法需要设置变量的变化范围，例如定义变量范围（如 W∈[0.13u, 20u]，L∈[0.13u, 10u]）。

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

提供基于 PyAether 的“**读取器件的连接关系**”的源代码，通过给定 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。

通过给定的“仿真引擎调用”PyAether 脚本，可以调用 MDE（模拟混合仿真环境）L2，通过集成在平台中的 ALPS 仿真器，例如可以调用给定的**单次仿真配置**：通过给定的全差分运放电路的仿真测试电路，进行电路仿真，得到开环 DC 增益，单位增益带宽，相位裕度，增益裕度和工作电流等参数。也可以调用给定 **PVT 条件的仿真配置**：包括 MOS 管，电阻和电容等随着工艺的变化，包括电源电压和环境温度的

变化等，PVT 条件进行电路仿真，得到开环 DC 增益，单位增益带宽，相位裕度，增益裕度和工作电流等参数。

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。

最终，参赛者可以通过**调用以下命令 run_opt_command.sh**，即可实现输入模拟 IC 的尺寸值文件（eg:extractcdfVal_0.txt），就可以调用 MDEL2 的设置，进行 ALPS 仿真（包含 PVT），得到仿真结果，即得到运放的开环直流增益 DC Gain，单位增益带宽 UGB，相位裕度 PM，增益裕度 GM 和工作电流 I_OPA 等仿真结果。

run_opt_command.sh 的命令如下：

```
pythone optimization.py
--ae_lib input_2025_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 extractcdfVal_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。/

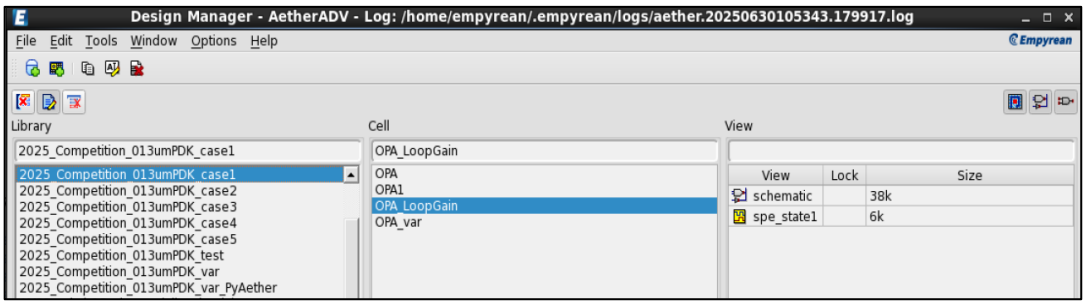
六、赛题解析

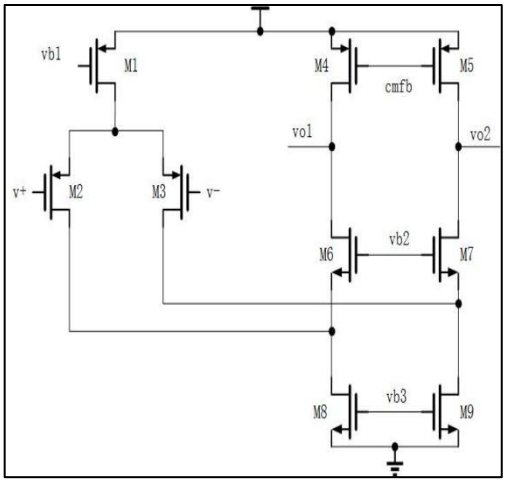
本赛题为基于 PyAether 的电路尺寸优化设计，在给定电路图架构和仿真设置下，通过算法优化满足目标参数。

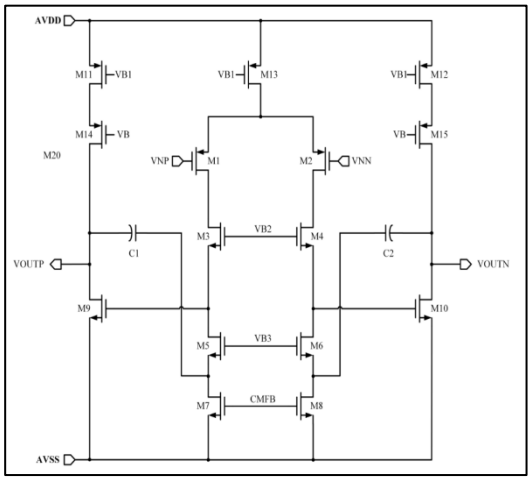
6.1 输入信息

1）电路设计原理图

全差分运放电路的电路图，包含主折叠运放/套筒运放和米勒补偿等电路。







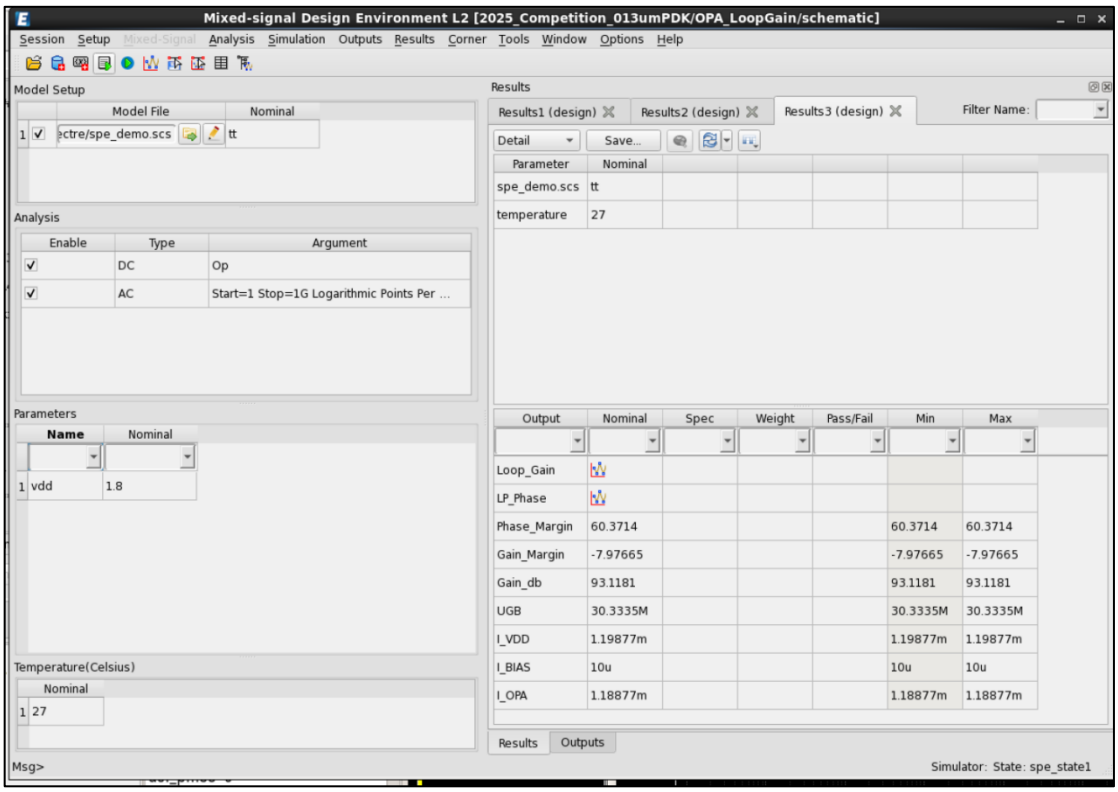
折叠式运放　　套筒式运放

2）电路仿真

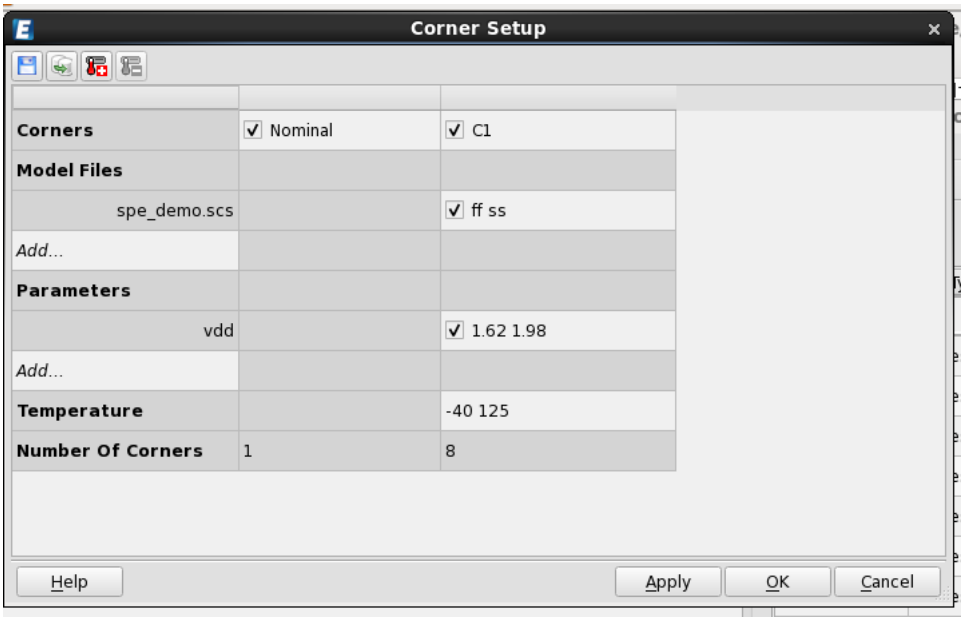
本赛题提供全差分运放电路的仿真测试电路图（含相关激励等）。

3）混合仿真环境设置

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。



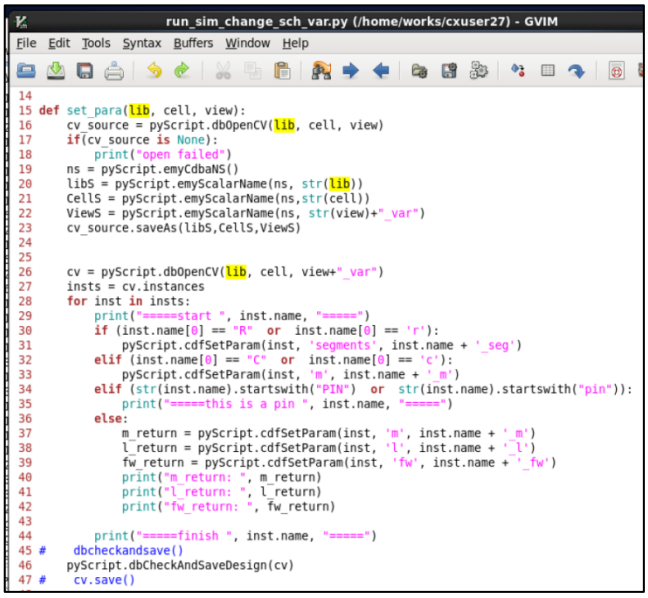
以及需要满足的 PVT 条件的设置，如下图所示：



，

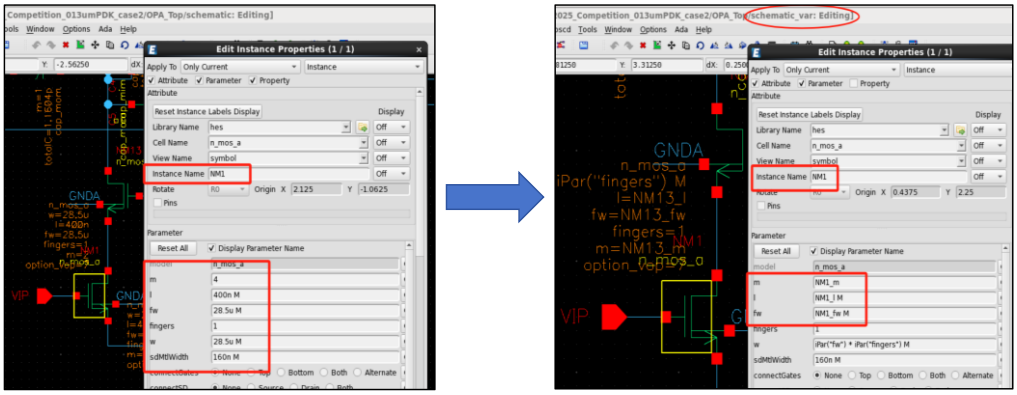
4）提供基于 PyAether 的“将电路图参数转变成变量”的源代码

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：




```
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。



schematic　　schematic_var

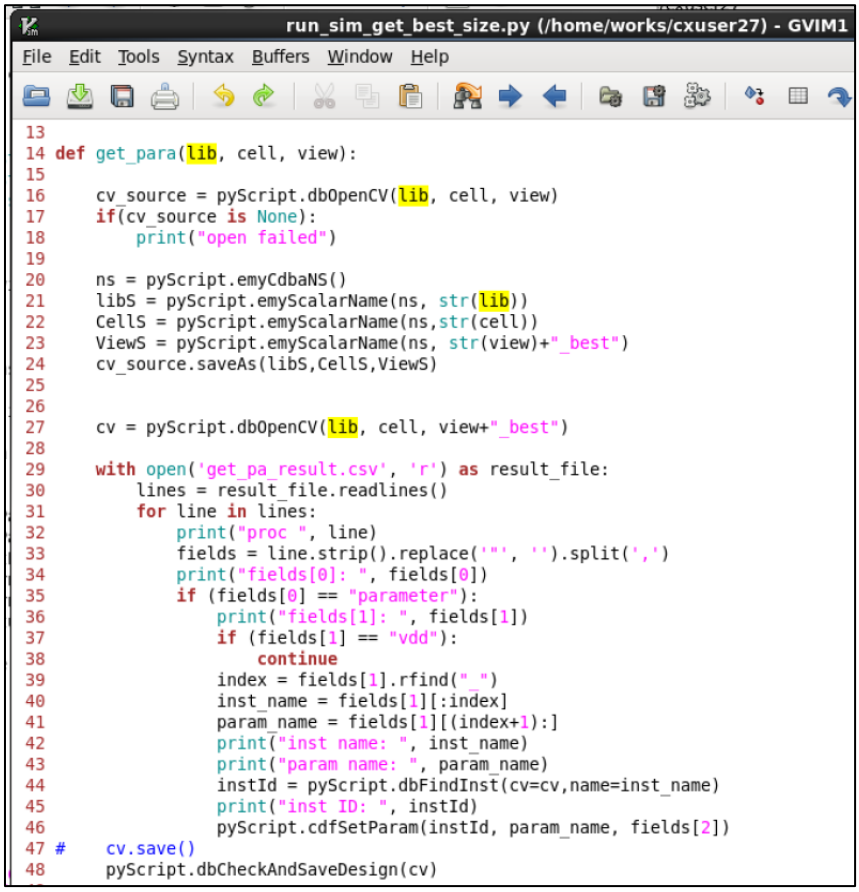
5）提供基于 PyAether 的“调用 MDE 进行仿真”的源代码

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
--------------------------------
```

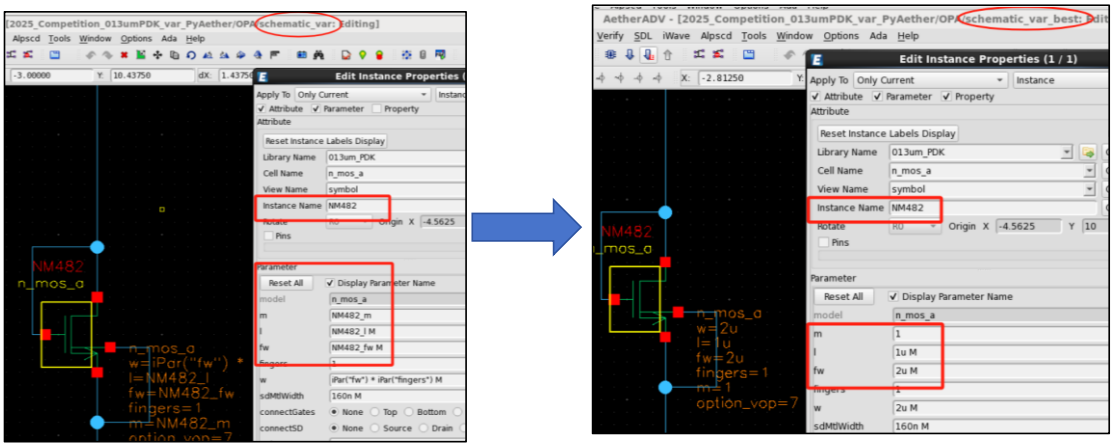
6）提供基于 PyAether 的“将最优电路尺寸数据反标到电路图”的源代码

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：




```
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。



schematic_var　　schematic_var_best

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：

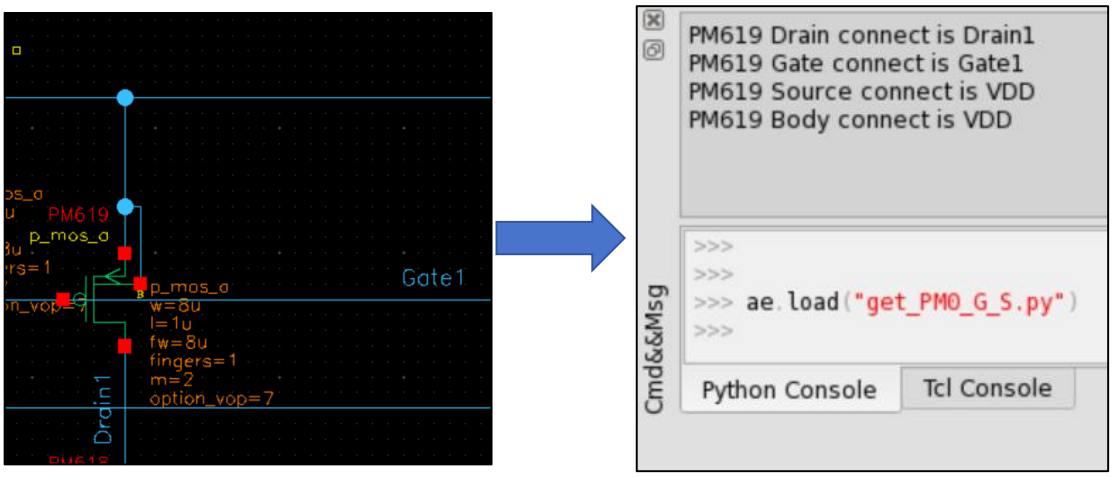
```
get_PM0_G_S.py = (/home/PyAether_Packages/Desktop/2025_EDA_Competition) - GVIM5
File  Edit  Tools  Syntax  Buffers  Window  Help

#open design
cv = ae.dbOpenCV(lib="2025_Competition_013umPDK_var_PyAether", cell="OPA", view="schematic")

I_PM0 = ae.dbFindInst(cv, "PM619")

for p in I_PM0.instTerms:
    if p.name=="G":
        print("PM619 Gate connect is", p.net.name)
    elif p.name =="S":
        print("PM619 Source connect is" ,p.net.name)
    elif p.name =="D":
        print("PM619 Drain connect is" ,p.net.name)
    else:
        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_2025_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 extractcdfVal_0.txt

其中：

蓝色部分为赛题输入部分，已经设置好； 绿色部分为仿真结果的输出结果，可以自行定义路径；红色部分为每次仿真所需输入的电路尺寸参数，是优算法所要输入的内容。注意：本赛题需要在给定的初始解条件下，进行优化。

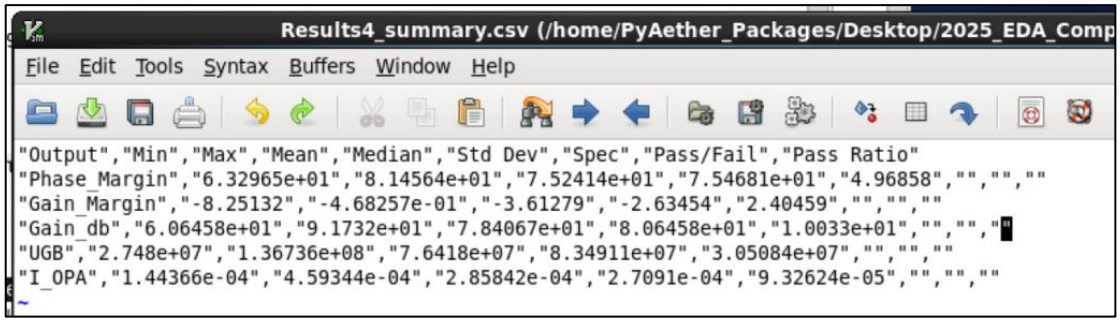
其运行如下所示：

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

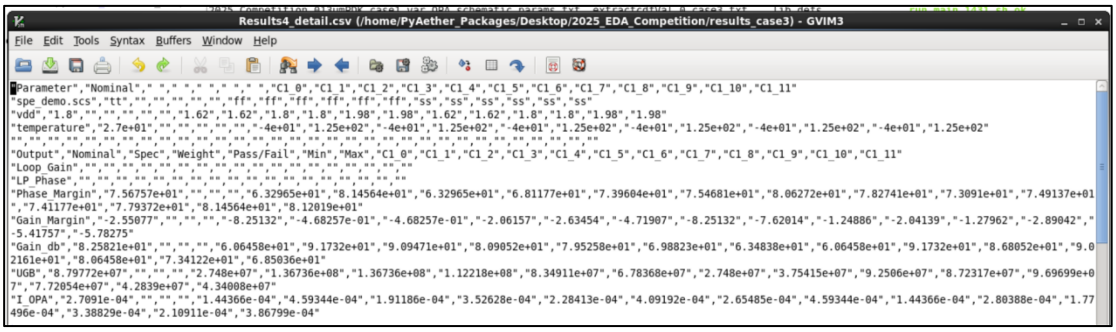
可以得到的仿真结果的文件如下所示：

```
[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：



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","ff","ff","ss","ss","ss","ss","ss","ss"
"vdd","1.8","","","","","","1.62","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","","","","","","-4e+01","1.25e+02","-4e+01","1.25e+02","-4e+01","1.25e+02","-4e+01","1.25e+02","-4e+01","1.25e+02","-4e+01","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","","","","","","","","","","","","","","","","","",""
"LP_Phase","","","","","","","","","","","","","","","","","",""
"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
","7.41177e+01","7.79372e+01","8.14564e+01","8.12019e+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","8.25821e+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","9.0
2161e+01","8.06458e+01","7.34122e+01","6.85036e+01"
"UGB","8.79772e+07","","","","2.748e+07","1.36736e+08","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+0
7","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","2.80388e-04","1.77
496e-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。

6.2 输出信息要求

1)器件识别之后的变量个数和最优的器件尺寸数值（如 M1_W=36u，M1_L=2u,R_W=2u）

2)最优尺寸下的反标后电路图（通过 PyAether 将尺寸反标到电路中,即 schmetic_var_best）

3)仿真验证报告（包含各 PVT 下的指标结果等），需在仿真报告中展示工艺 TT/FF/SS，高温（125℃）、低温（-40℃），和标称电压±10%变化的仿真结果

4)优化运行次数，运行时间和最终结果

七、上机环境

1. 远程登录 VNC，Centos6.9 的服务器环境。

2. 配置好的 Pycharm。

3. 参赛者需要通过使用 Python 等编程方法实现本赛题。

4. 相关工具：ALPS 仿真器，Aether 全定制设计平台，PyAether 生态系统等。

八、评分标准

A. 提交要求

参赛者需要通过使用 Python/C++等编程语言，结合 PyAether 接口来实现本赛题。

1. 代码：Python/C++ 源码，包含参数化电路图生成、仿真流程、优化算法；
2. 文档：设计架构、算法原理、优化结果对比（含电路图截图、仿真曲线）、测试报告；
3. 输出文件：优化后的网表、仿真结果等评估数据。

需要提交源代码、设计文档（包含架构、算法、生成的电路图截图，测试报告）等相关信息。

B. 打分器

给定样本分为给定已知运放 Public 电路（2 个）和内部 Hidden 测试运放电路（2 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。

本赛题要求输出报告，主要包括需要满足输出信息 6.2 章节的内容，包括但是不限于通过给定的 PyAether 的相关源代码，和已知电路图，生成最佳尺寸的电路图，和仿真结果等；具体见附件的报告模板。

注意：必须按照模板，提交本输出报告。

序号	输出内容	格式要求	评价重点
1	器件识别之后的变量个数和最优的器件尺寸数值	CSV 文件（如 M1_w=36u, M1_L=2u）	满足相关约束

序号	输出内容	格式要求	评价重点
2	反标后电路图	.schematic_var_best（含最优尺寸）	尺寸可制造性
3	仿真验证报告	最优尺寸下的仿真结果和波形（含 PVT 结果）	性能达标率与鲁棒性
4	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	综合优化能力

(1) 【40 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 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
直流增 DC Gain≥ 80dB	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

(2) 【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
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	25 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 fw*L*m*1.5 ● 电容面积为： W*L*1 ● 电阻面积为：W*L*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。

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*15。

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*25。

(3) 【20 分】总体运行总时间，0~20 分。

基础得分：以满足至少 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*20，最高基础得分

为 20 分。

每当有一项约束条件不满足，则本项扣 5 分。

九、参考文献

[1]. Hald, J., & Madsen, K. (n.d.). A 2-STAGE ALGORITHM FOR MINIMAX OPTIMIZATION. Technical University of Denmark, Lyngby, Denmark.

[2]. P.C. Maulik, L.R. Carley, "Automating analog circuit design using constrained optimization techniques," IEEE International Conference on Computer-Aided Design Digest of Technical Papers, Santa Clara, CA, USA, 1991.

[3]. Markus Leibl, Abdelaziz Lberni, Helmut Graeb, "On the Importance of Initial Sizing of Analog Circuits Based on Analytical Equations," 2024 31st IEEE International Conference on Electronics, Circuits and Systems (ICECS), Nancy, France, 2024.

[4]. Ria Rashid, Nandakumar Nambath, "Area Optimisation of Two Stage Miller Compensated Op-Amp in 65 nm Using Hybrid PSO," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 69, no. 1， 2022.

[5]. Rahul Dutta, Ashish James, Salahuddin Raju, Yong-Joon Jeon, Chuan Sheng Foo, Kevin Tshun Chuan Chai, "Automated Deep Learning Platform for Accelerated Analog Circuit Design," IEEE 35th International System-on-Chip Conference, Belfast, United Kingdom, 2022.

[6]. Abdelaziz Lberni, Malika Alami Marktani, Abdelaziz Ahaitouf, Ali Ahaitouf, "Adaptation of the Whale Optimization Algorithm to the Optimal Sizing of Analog Integrated Circuit: Low Voltage Amplifier Performances," 2020 IEEE 2nd International Conference on

Electronics, Control, Optimization and Computer Science (ICECOCS), Kenitra, Morocco, 2020.

*本赛题指南未尽问题，见赛题 Q&A 文件

2025 China Postgraduate IC Innovation Competition · EDA Elite Challenge Contest

1. Problem

Automated Sizing Optimization for Analog Circuit Design --- Co-creating the Future of EDA with PyAether Ecosystem

2. Company

Beijing Empyrean Technology Co., Ltd.

3. Required Knowledge Background

3.1 No specific major background is required. The core task is to perform automated sizing optimization for analog circuits using the PyAether full-custom design platform ecosystem.

3.2 Suitable for students in computer science, microelectronics, electronic information engineering, automation, mathematics, etc.

3.3 Participants should have programming experience with Python, C, C++, or Matlab.

4. Problem Background

4.1 Background of Analog Circuit Design Automation

In modern IC (integrated circuit) design, analog circuit design is complex and time-consuming, especially for high-performance circuits like fully differential high-gain amplifiers. Traditional manual methods rely on engineers' experience, and manual optimization becomes increasingly difficult, as they are inefficient and error-prone. Thus,

implementing automated sizing optimization for analog circuits is crucial for improving design efficiency and shortening development cycles.

4.2 Introduction of Fully Differential High-Gain Amplifier

Fully differential operational amplifiers (op-amps) serve as fundamental building blocks in analog circuits, finding extensive applications in data converters, filters, sensors, and signal processing systems. Their distinguishing features include high gain and excellent common-mode rejection ratio (CMRR). Key performance specifications encompass: Open-loop gain (DC Gain, typically ≥100dB) for noise and distortion suppression; Phase margin (PM, typically ≥50°) and gain margin (GM, typically ≤-6dB) to ensure system stability; Power supply rejection ratio (PSRR @DC≥112dB and ≥80dB at 1MHz) for power noise immunity; Power consumption specifications (e.g., operating current Isum ≤1mA, standby current Istandby_current≤100nA) to meet low-power requirements. These amplifiers typically achieve their high gain and stability through multistage amplification and Miller compensation techniques. However, manual optimization requires careful consideration of multiple parameters, including: Transistor width-to-length ratios (W/L)、Bias current settings、Compensation network design. This optimization process often necessitates numerous iterative attempts to achieve optimal performance.

4.3 Significance and Necessity of Automated Sizing Optimization

Automated sizing optimization is a critical step in analog circuit design that can significantly enhance design efficiency, reduce development cycles, and improve circuit performance consistency. Through automated sizing algorithms, designers can explore numerous design

variables within a short time to identify optimal device sizing parameter combinations that meet specifications including gain, bandwidth, and power supply rejection ratio (PSRR).

- **Efficiency Improvement**: Replaces manual trial-and-error with algorithms to shorten design cycles;
- **Performance Optimization:** Employs multi-objective optimization (gain, bandwidth, stability, area, etc.) to achieve globally optimal solutions;
- **PVT Compatibility:** Automatic parameter adjustment for varying PVT (Process, Voltage, Temperature) conditions to ensure design reliability.

5. Competition Problem Description

This competition aims to evaluate automated sizing algorithms for analog circuits. Given circuit schematics, simulation setups, and PyAether source code for automation support, participants must develop optimization algorithms to achieve automated sizing of analog circuits while meeting constraints and optimizing op-amp performance—thereby advancing analog design automation to improve efficiency and reduce development cycles.

5.1 Task: Multi-Objective Sizing Optimization Algorithm Development Under Constraints

Task: Identify device structures to reduce design variables; develop optimization algorithms to determine optimal sizing that meets constraints and optimization targets.

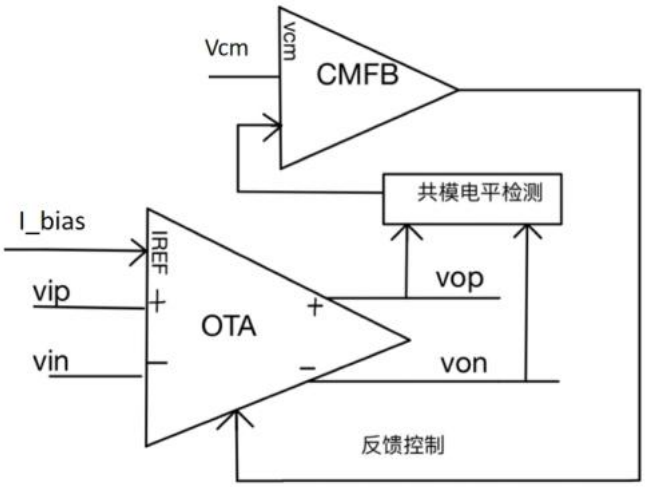
- **Optimization Variables:** Transistor width, length (W， L), resistor/capacitor values, etc.
- **Constraints:** Phase margin (PM), gain margin (GM), and DC Open Loop gain, and operating current (I_OPA) must meet specified thresholds across PVT conditions. For details, see Chapter 8 of the scoring section.

- **Optimization Targets:** unity-gain bandwidth (UGB), operating current (I_OPA), and preliminary area estimation. The unity-gain bandwidth should be as large as possible, and the area should be as small as possible. For details, see Chapter 8 of the scoring section.
- **Device Identification and Variable Number Optimization:** Based on PyAether, identify matched structures in analog ICs (e.g., current mirrors, input pairs, active loads, matched resistors, capacitor arrays, dummy devices, etc.) to significantly reduce the number of optimization variables and improve optimization efficiency. The post-identification variable count must be obtained.
- **Algorithm Types:** Include but are not limited to differential evolution, genetic algorithms, simulated annealing, Bayesian optimization, AI (machine learning algorithms), etc. Novel self-developed algorithms are also welcomed, to finally get determined transistor dimensions, resistor and capacitor values, etc.
- **Output Results:** The count of reduced variables, and the optimal parameter sizes obtained through optimization algorithms. Based on provided PyAether script examples, output the optimized circuit schematic along with PVT simulation results under the optimal dimensions.

5.2 Test Cases to be Provided to Participating Teams: Including Circuit Design and Simulation Settings, etc.

Test cases will be provided to participating teams, including multiple fully differential operational amplifieranalog circuit diagrams. These circuit diagrams include core operational amplifiers (such as folded operational amplifiers, telescopic operational amplifiers, as well

as common-mode feedback circuits), circuit simulation test diagrams (Test Bench) including excitations, and analog mixed simulation environment (MDE). MDE mainly includes single simulation design environment and output results (gain and phase, unity gain bandwidth, phase margin, gain margin, operating current, etc.), as well as settings of PVT conditions to be met, etc.



5.3 Relevant Scripts for Auxiliary Automation Process Based on PyAether to be Provided to Participating Teams

- Relevant scripts for the auxiliary automation process based on PyAether will be provided to participating teams, mainly including:
- Parameterized PyAether scripts for converting device sizes in schematic into variables to be optimized
- PyAether scripts for reading the connection relationships of devices
- PyAether scripts for simulation engine invocation that can call MDE for PVT simulation and output performance targets

- Back-annotation PyAether scripts for finally writing the optimal size into the circuit diagram
- More integrated PyAether scripts for automatic optimization interface where participants only need to input the size file to call the simulation and obtain the simulation results of this size

Through "**the parameterized PyAether scripts for converting device sizes into variables**", devices in the original circuit can be converted into variables, such as transistor width-to-length ratio (W/L), resistance value, capacitance value, etc. (e.g., M1_W=16u, Rc=10k). Therefore, the optimization algorithm needs to set the variation range of variables, such as defining variable ranges (e.g., W∈[0.13u, 20u], L∈[0.13u, 10u]).

Device Type	Parameter	Value Range
Transistor	finger width value	0.13 μm – 20 μm
	L value	0.13 μm – 10 μm
	Multiplier(m)	1,2,....20(integer)
Resistor	Segment	1,2,....20(integer)
Capacitor	L value	0.13 μm – 20 μm

The source code for "**reading device connection relationships**" based on PyAether is provided. Through the given PyAether code, the connection relationships of devices in the circuit diagram can be read, and the names of Nets connected to the device's Gate, Source, etc. can be automatically printed.

Through the given PyAether scripts for "**simulation engine invocation**", MDE (analog mixed simulation environment) L2 can be called, and through the ALPS simulator integrated in the platform, for example, the given **single simulation configuration** can be invoked: circuit simulation is performed through the given simulation test circuit of the fully differential operational amplifier to obtain parameters such as open-loop DC gain, unity gain bandwidth, phase margin, gain margin, and operating current. The simulation configuration with given PVT conditions can also be invoked: circuit simulation under **PVT conditions** (including changes in MOS transistors, resistors, capacitors, etc. with process, power supply voltage, ambient temperature, etc.) is performed to obtain parameters such as open-loop DC gain, unity gain bandwidth, phase margin, gain margin, and operating current.

The source code for "**back-annotating optimal circuit size data to the circuit diagram**" based on PyAether is provided. By running this script with PyAether and the csv file of optimal size parameters obtained by the optimization algorithm in this assessment, the circuit diagram with optimal size can be automatically generated.

Finally, participants can **call the following command run_opt_command.sh** to input the size value file of the analog IC (e.g., extractcdfVal_0.txt), then MDEL2 settings can be invoked to perform ALPS simulation (including PVT) and obtain simulation results, i.e., simulation results such as open-loop DC gain (DC Gain), unity gain bandwidth (UGB), phase margin (PM), gain margin (GM), and operating current (I_OPA) of the operational amplifier.

The command for run_opt_command.sh is as follows:

```
pythone optimization.py
--ae_lib input_2025_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 extractcdfVal_0.txt

Wherein:

The blue part is the input part of the problem, which has been set:

Input information, the Lib name of the analog circuit to be optimized, which has been set

Input information, the Cell name of the analog circuit to be optimized, which has been set

Input information, the View name of the analog circuit to be optimized, which has been set

Input information, the TestBench of the circuit to be optimized, which has been set

Input information, MDE L2 corresponding to the TestBench of the circuit to be optimized, which has been set

The green part is the output result of the simulation, and the path of the output result can be defined by oneself:

Output path of simulation results

Output summary of all simulation results (gain, margin, UGB, etc.)

The red part is the circuit size parameters to be input for each simulation.

Input file, here is the input initial circuit size parameter file: extractcdfVal_0.txt; participants need to obtain a new circuit size file according to the optimization algorithm they use to replace this .TXT file, so as to obtain new simulation results.

Note: This problem needs to be optimized under the given initial solution conditions.

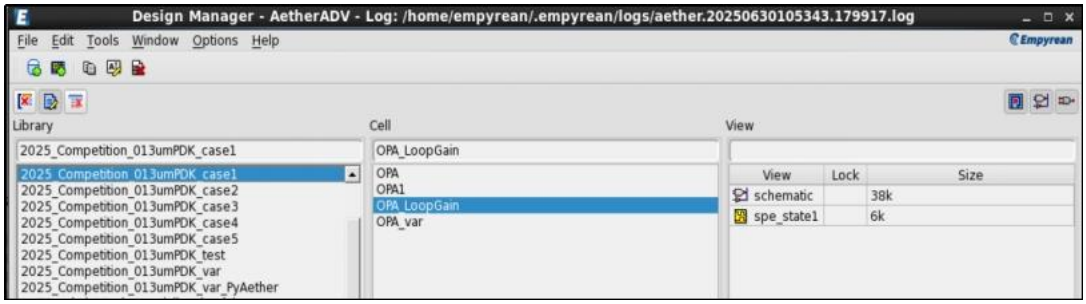
6. Problem Analysis

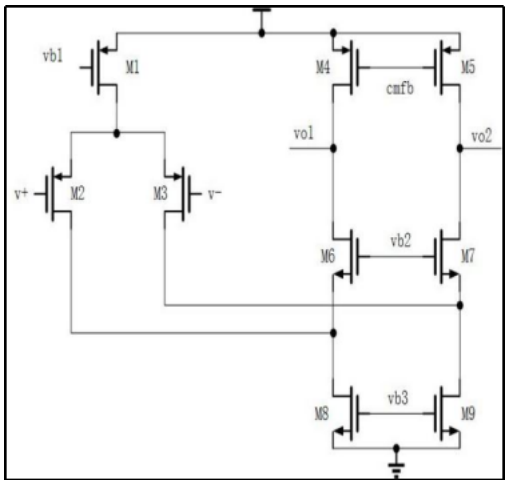
This test is about circuit size optimization design based on PyAether. Under the given circuit diagram architecture and simulation settings, the target parameters are met through algorithm optimization.

6.1 Input Information

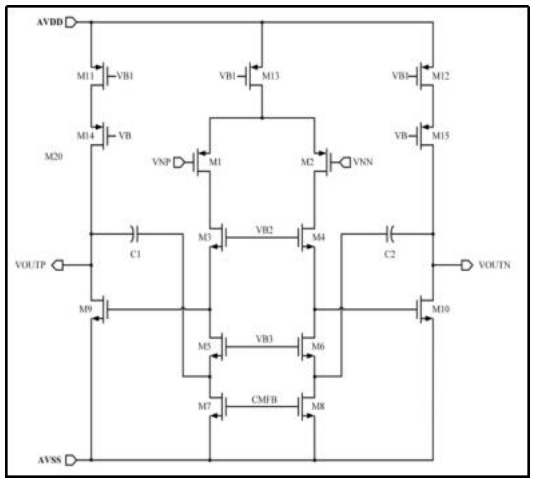
1) Circuit Design Schematic

The circuit diagram of the fully differential operational amplifier includes main folded operational amplifiers/telescopic operational amplifiers and Miller compensation circuits, etc.





Folded Amplifier



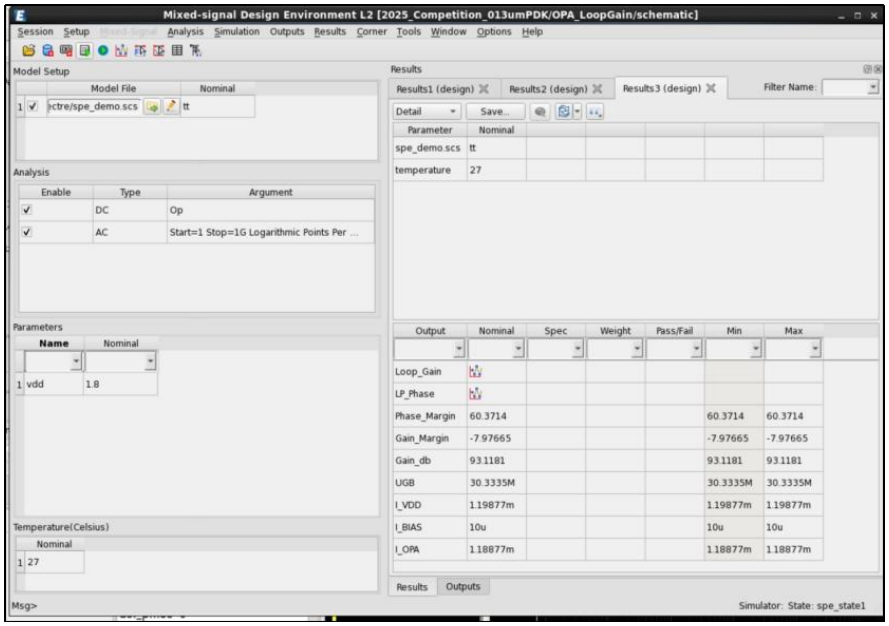
Telescopic Operational Amplifier

2) Circuit Simulation Test Bench

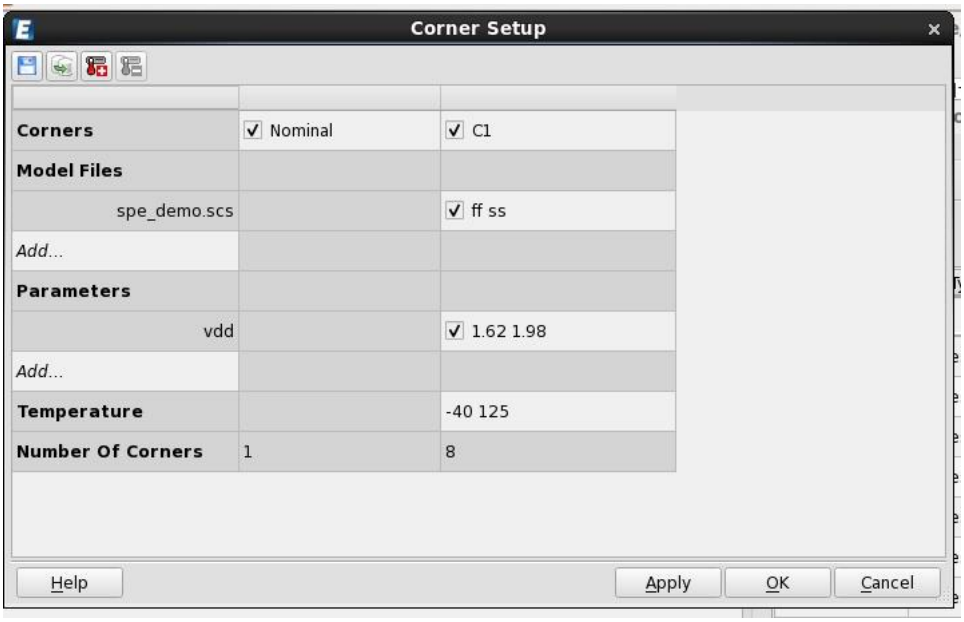
Here provides the simulation test bench of the fully differential operational amplifier (including relevant stimulus, etc.).

3) Mixed Simulation Environment Settings

Including single simulation design environment and output results, etc. The output results include Gain and Phase, unity gain bandwidth（UGB）, phase margin(PM）, gain margin（GM）, operating current（I_OPA）, etc.

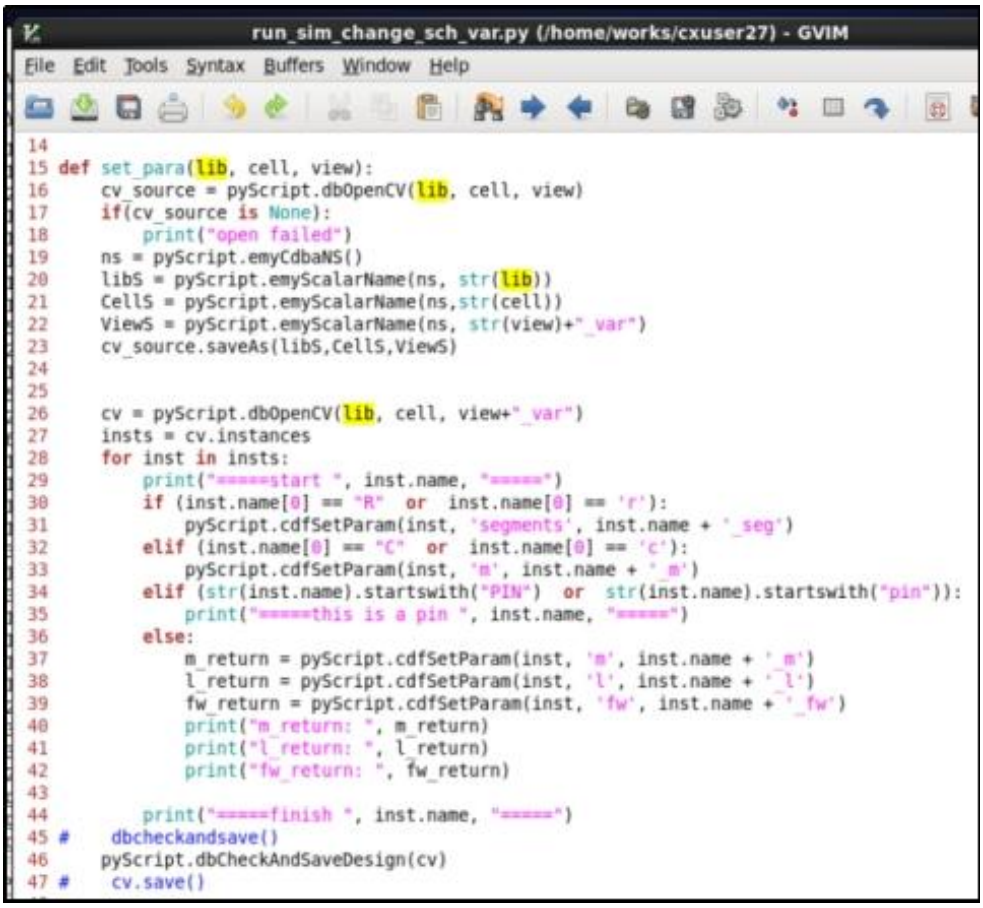


And the settings of PVT conditions to be met, as shown in the figure below:



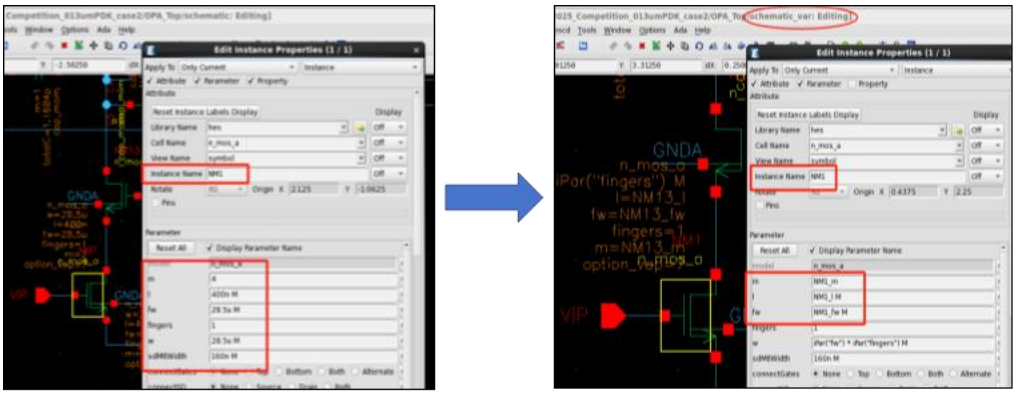
4) Provide Source Code for "Converting Circuit Diagram Parameters to Variables" Based on PyAether

By running this script with PyAether and the known circuit diagram, a parameterized circuit diagram can be automatically generated, that is, the sizes of transistors, resistors, and capacitors in the known circuit diagram are converted into variables. The core code is as follows:




```
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()
```

After running this PyAether script, a schematic_var circuit diagram with various variables can be generated. These variables are mainly formed by adding _cdf parameters to the instance names, such as NM1_m, etc.



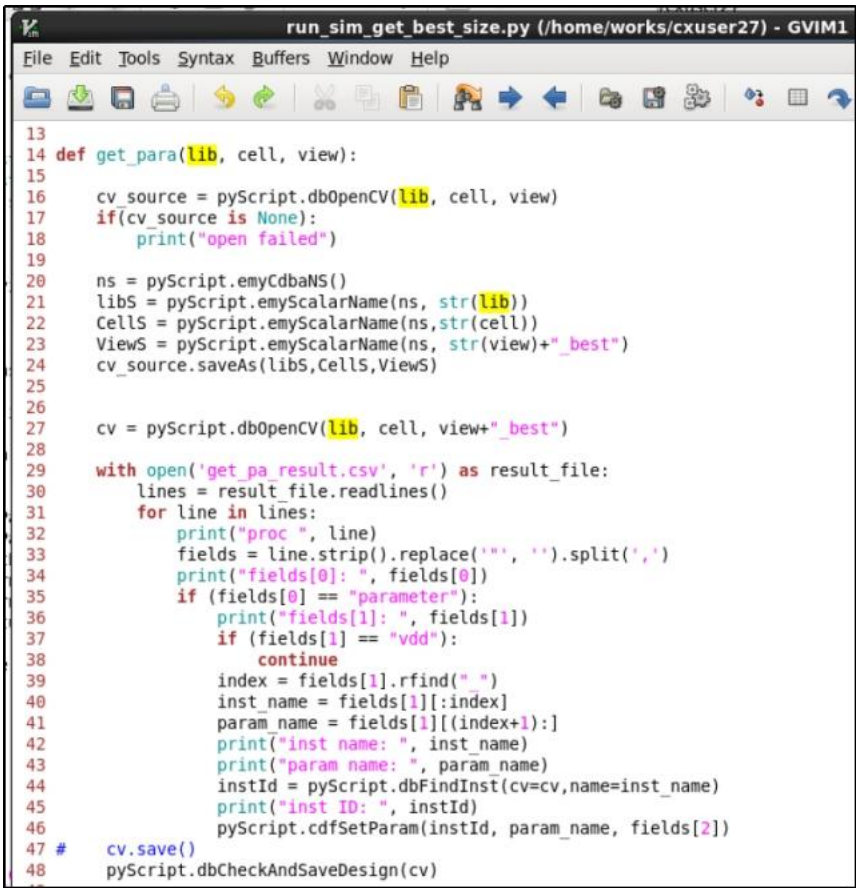
schematic　　schematic_var

5) Provide Source Code for "Calling MDE for Simulation" Based on PyAether

By running this script with PyAether and the known MDE, the corresponding simulation can be automatically called, and the simulation results can be obtained. The running effect is as follows:

```
------------------------------
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
------------------------------
```

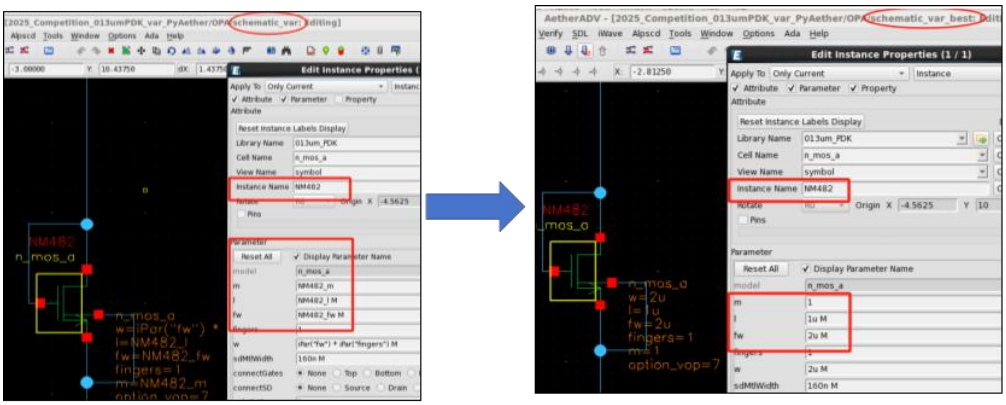
6) Provide Source Code for "Back-annotating Optimal Circuit Size Data to the Circuit Diagram" Based on PyAether




```
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)
```

By running this script with PyAether and the csv file of optimal size parameters obtained by the optimization algorithm in this assessment, the circuit diagram with optimal size can be automatically generated. The core code is as follows:

Then it will generate a schematic_var_best schematic containing the optimal parameters.



schematic_var　　schematic_var_best

7) Provide Source Code for "Reading Device Connection Relationships" Based on PyAether

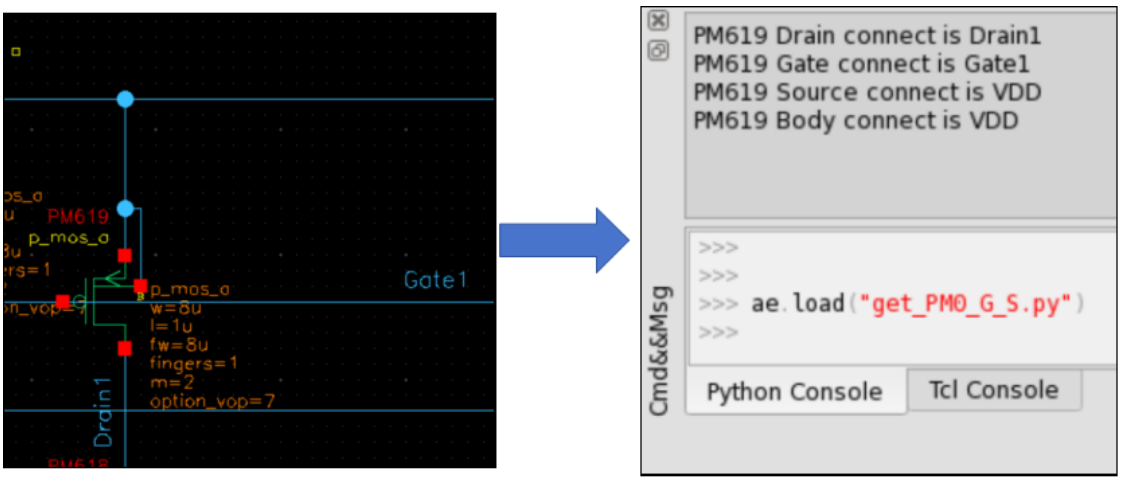
Through the given PyAether code, the connection relationships of devices in the circuit diagram can be read, and the names of Nets connected to the device's Gate, Source, etc. can be automatically printed. The core code is as follows:

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)
```

The running effect is as follows:



schematic 读取的连接关系

8) Provide Overall Source Code for Automatic Optimization Interface Based on PyAether Where Participants Only Need to Input Size Files

Finally, participants can call the following command run_opt_command.sh to run the simulation under the corresponding size and obtain the simulation results.

The command for run_opt_command.sh is as follows:

pythone optimization.py

--ae_lib input_2025_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 extractcdfVal_0.txt

Wherein:

The blue part is the pre-configured input part of the problem, which has been set up; the green part is the output result of the simulation, and the path can be defined by oneself; the red part is the circuit size parameters to be input for each simulation, which are the input contents of the optimization algorithm. Note: This problem needs to be optimized under the given initial solution conditions.

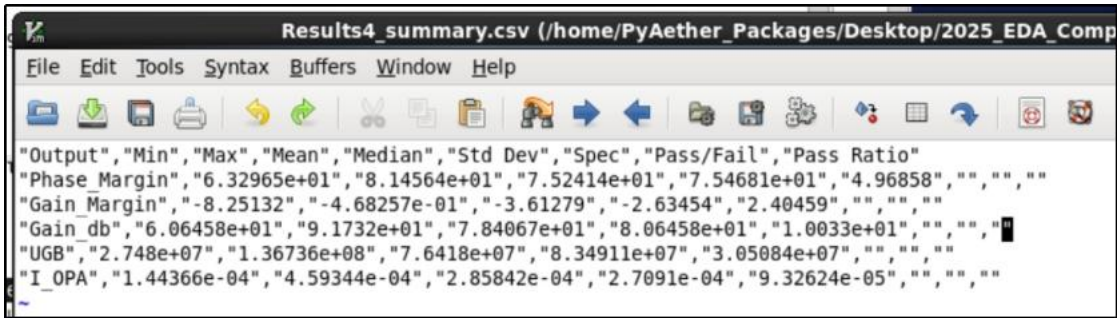
Its operation is as follows:

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

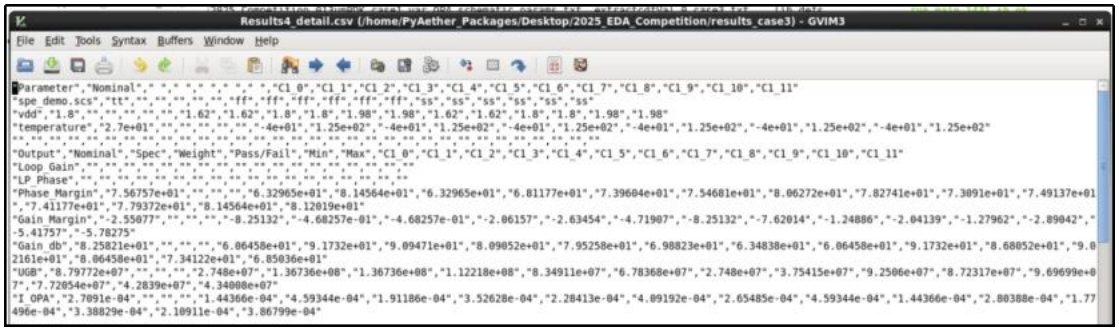
The available simulation result files are as follows:

```
[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
```

The summary of simulation results can be viewed as follows:



The specific results of each simulation can be viewed:



9) Provide Process Constraints:

Relevant process PDK libraries are provided.

6.2 Output Information Requirements

1) The number of variables after device identification and optimal device size values (e.g., M1_W=36u, M1_L=2u, R_W=2u).

2) Back-annotated circuit diagram with optimal size (size is back-annotated to the circuit through PyAether, i.e., schematic_var_best).

3) Simulation verification report (including index results under various PVT conditions, etc.), which needs to show simulation results of processes TT/FF/SS, high temperature (125℃), low temperature (-40℃), and nominal voltage ±10% variation in the simulation report.

4) Number of optimization runs, running time, and final results.

7. Computer Environment

6.1 Remote login to VNC, server environment of Centos6.9

6.2 Configured Pycharm

6.3 Participants need to implement this problem using programming methods such as Python

6.4 Relevant tools: ALPS simulator, Aether full-custom design platform, PyAether ecosystem, etc.

8. Scoring Criteria

A. Submission Requirements

Participants need to implement this problem using programming languages such as Python/C++ combined with PyAether interfaces.

1. Code: Python/C++ source code, including parameterized circuit diagram generation, simulation process, and optimization algorithm;
2. Documents: Design architecture, algorithm principle, comparison of optimization results (including circuit diagram screenshots, simulation curves), test report;
3. Output files: Optimized netlist, simulation results, and other evaluation data.

Relevant information such as source code, design documents (including architecture, algorithm, screenshots of generated circuit diagrams, test reports) need to be submitted.

B. Scoring Mechanism

The given samples consist of public known two op-amp circuits (2) and three hidden test op-amp circuits (3). The public op-amp circuits are disclosed to participants for algorithm development verification, while the hidden test op-amp circuits are kept confidential to evaluate the generality of participants' algorithms. The public circuits are not included in scoring, while the hidden test op-amps are used for evaluation. The maximum runtime for each op-amp circuit's complete code execution is 4 hours.

This competition requires submission of a report containing the output information specified in Section 6.2, including but not limited to: generating optimized circuit schematics and simulation results using the provided PyAether source code and known circuit diagrams. Please refer to the attached report template for details.

Note: The output report must be submitted according to the template.

Serial Number	Output Content	Format Requirement	Evaluation Focus

1	Number of variables after device identification and optimal device size values	CSV file (e.g., M1_w=36u, M1_L=2u)	Meeting relevant constraints
2	Back-annotated circuit diagram	.schematic_var_best （including optimal size）	Manufacture ability
3	Simulation and verification report	Simulation results and waveform under optimal size (including PVT results)	Performance compliance rate and robustness
4	Code quality and algorithm performance	Including number of optimization runs, running time, memory usage, etc.	Comprehensive optimization capability

(1) [40 points] Satisfaction of constraints based on the given circuit

Through the given circuit diagram, MDE environment, and PyAether source code, the most basic constraint conditions for phase margin (PM), gain margin (GM), DC gain (DC Gain), and operating current (I_OPA) are met. If any constraint is not met, such as phase margin not meeting the requirement, 10 points will be automatically deducted, so the score for this item is 0. If met, full marks will be given directly.

Constraints	Score	**Constraint Condition**
Phase Margin，PM≥50°	10 points	Must be met for all PVT

Gain Margin， GM≤-10dB	10 points	Must be met for all PVT
DC Loop Gain， DC Gain≥ 80dB	10 points	Must be met for all PVT
Operating Current I_OPA≤3mA	10 points	Must be met for all PVT

(2) [40 points] Evaluation of Optimization Objectives for Multi-Objective Sizing Optimization Algorithm

Comprehensively considering targets such as unity gain bandwidth (UGB) and preliminary area sum, the scorer will evaluate the quality of the final results based on the following indicators:

Objective	Score	Evaluation Condition
Unity Gain Bandwidth (UGB) The larger the better	15 points	Worst-case scenario for all PVT corners
Area minimization	25 points	Note: The area parameter is initially calculated as the sum of the areas of all transistors, resistors, and capacitors. Among them: ● the area of each transistor is fwLm1.5 ● the area of the capacitor is: WL1

		● the area of the resistor is: WL*2

Each objective is scored according to the ranking: the first place gets full marks for this item; based on the first place, other scores are calculated proportionally.

Taking the indicator of the team with the largest UGB as UGB0, and the UGB indicator of other teams as UGB1, the UGB score is UGB1/UGB0*15.

Taking the team with the smallest area as full marks, with the smallest area recorded as A0, and the area of other teams, e.g., A1, the area score is A0/A1*25.

(3) [20 points] Total running time, 0~20 points.

Basic score: Taking the shortest running time among teams that meet at least 3 constraint conditions as T0, and the running time of other teams as T1, the basic score for the running time of the team is T0/T120, with the maximum basic score being 20 points.

For each unmet constraint condition, 5 points will be deducted from this item.

9. References

[1] Hald, J., & Madsen, K. (n.d.). A 2-STAGE ALGORITHM FOR MINIMAX OPTIMIZATION. Technical University of Denmark, Lyngby, Denmark.

[2] P.C. Maulik, L.R. Carley, "Automating analog circuit design using constrained optimization techniques," IEEE International Conference on Computer-Aided Design Digest of Technical Papers, Santa Clara, CA, USA, 1991.

[3] Markus Leibl, Abdelaziz Lberni, Helmut Graeb, "On the Importance of Initial Sizing of Analog Circuits Based on Analytical Equations," 2024 31st IEEE International Conference on Electronics, Circuits and Systems (ICECS), Nancy, France, 2024.

[4] Ria Rashid, Nandakumar Nambath, "Area Optimisation of Two Stage Miller Compensated Op-Amp in 65 nm Using Hybrid PSO," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 69, no. 1， 2022.

[5] Rahul Dutta, Ashish James, Salahuddin Raju, Yong-Joon Jeon, Chuan Sheng Foo, Kevin Tshun Chuan Chai, "Automated Deep Learning Platform for Accelerated Analog Circuit Design," IEEE 35th International System-on-Chip Conference, Belfast, United Kingdom, 2022.

[6] Abdelaziz Lberni, Malika Alami Marktani, Abdelaziz Ahaitouf, Ali Ahaitouf, "Adaptation of the Whale Optimization Algorithm to the Optimal Sizing of Analog Integrated Circuit: Low Voltage Amplifier Performances," 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), Kenitra, Morocco, 2020.

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