

README

本文介绍了2024年EDA精英挑战赛赛题十《针对模拟电路的电路图到网表自动生成》的测试平台使用方法。参赛队伍可以选择（1）在服务器上运行测试代码，也可以选择（2）将本文件夹中的测试文件下载到本地，完成本地环境的配置，并按照下述的使用方法完成测试。

文件说明

- images/ 文件夹，储存测试图片。命名方法为 序号.png 如 001.png
- true/ 文件夹，存放正确网表，命名方法为 序号.txt 如 001.txt
- generate/ 文件夹，需要自建，存放参赛队伍生成的网表，命名方法同 true 文件夹
- [main.py](#) 测试代码
- [utils.py](#) 测试代码

运行 `python main.py`，测试代码会遍历generate文件夹中的所有文件，寻找所有的true文件夹中与其同名的文件，将两者进行比较。注意，generate文件夹中参赛队员生成的文件格式为python的dict字符串格式，并非json格式。如用户在python代码中将电路图转换为一个字典结构的网表 `netlist`，则在文件中使用 `f.write(str(netlist))` 写入。

环境依赖

服务器上的运行环境已为大家配置好了，若选择（1）在服务器上运行代码，只需在运行时选择 Python3.8.10 64-bit (路径为/bin/python3)的python解释器即可，并将codes文件夹中需要拷贝的文件拷贝到用户自己的文件夹中；若选择（2）在本地完成测试，则需要自行完成环境配置：

```
python 3.10+
networkx 3.1
numpy 1.24.4
torch 2.4.0
pydantic 2.9.1
pandas 2.0.3
dgl 2.4.0
```

建议使用conda 安装。conda安装方式：[conda](#)

然后执行

```
conda create -n val_circuit_img python=3.12
conda activate val_circuit_img
pip install networkx==3.1 numpy==1.24.4 pydantic==2.9.1 pandas==2.0.3
# 安装torch与dgl，以下为示例
pip install torch==2.4.0
pip install dgl -f https://data.dgl.ai/wheels/torch-2.4/repo.html
```

dgl安装方式参考：[dgl](#)

运行

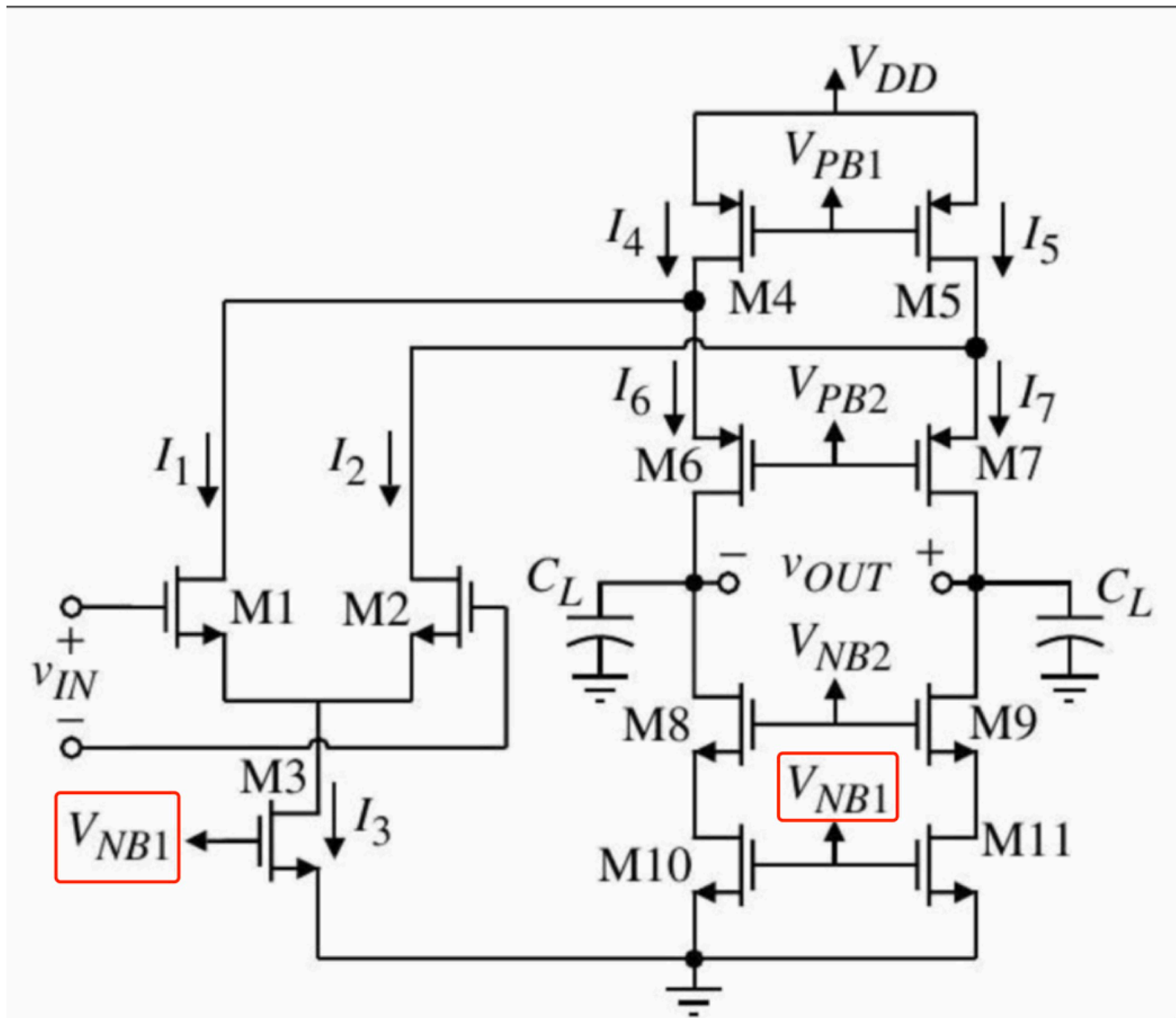
1. 编写自己的识别代码，将输入目录设为 `image/`，输出目录设置为 `generate/`
2. 运行自己的识别代码，将图片 `image/001.png` 的识别结果存放在 `generate/001.txt` 中，以此类推（在参赛队本地测试自己的代码效果时，如果要用服务器上的测试环境，也可以手动拷贝图片到服务器上的generate文件夹）
3. 切换到测试代码环境，运行 `python main.py`
4. 获得测试报告 `validation_report.md` 在测试代码根目录

打开`validation_report.md`文件，可以查看测试报告。报告列出了20个测例的测试结果以及综合测试结果。关于功能识别精确度F、网表识别精确度K和运行时间的定义和计算方法，详见[赛题指南](#)。

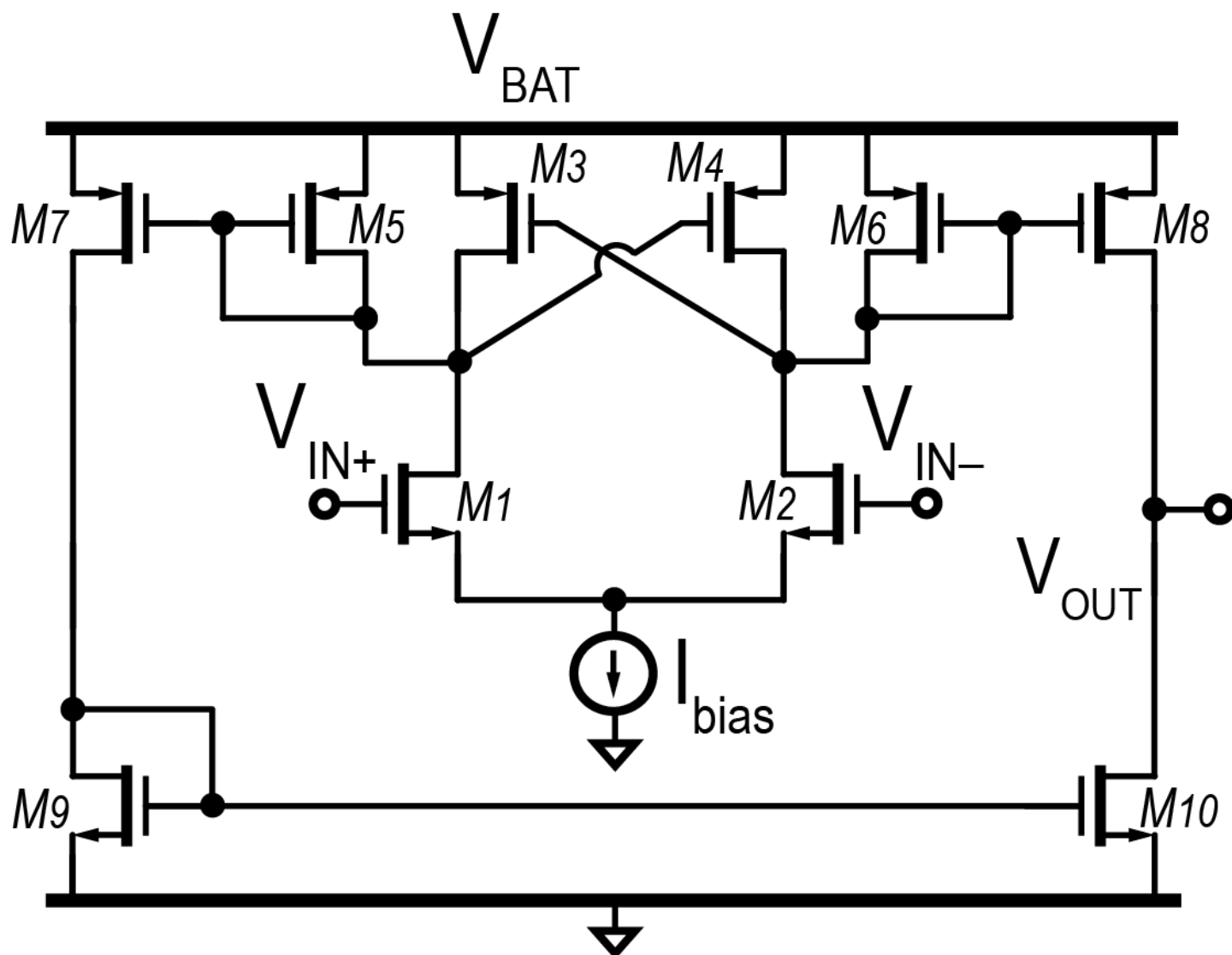
部分规则的补充说明

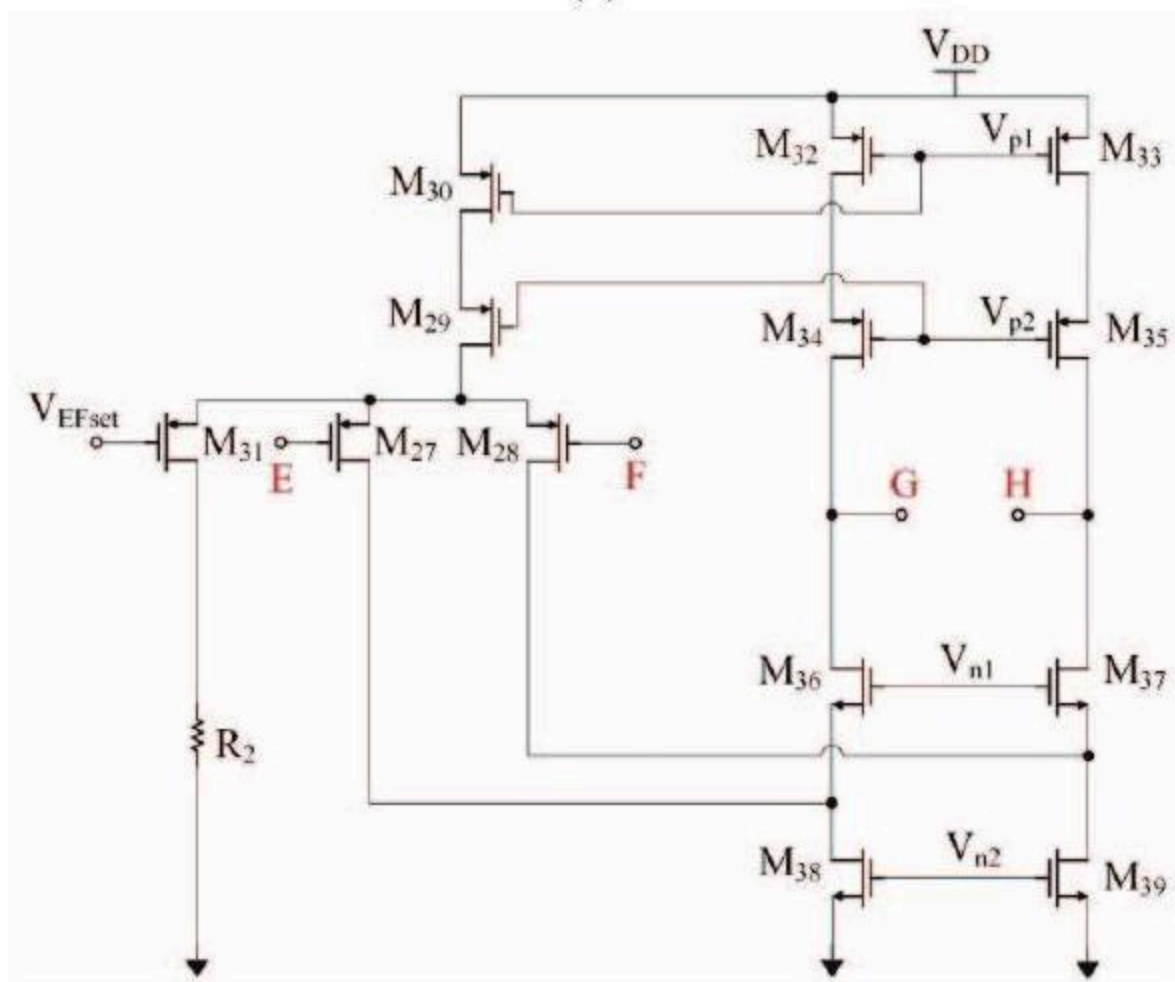
VDD和GND

注意，我们不考虑文字识别。换句话说，即便图中存在两个命名相同的非连接网络，我们依然认为他们是两个完全不同的网络。如图（1）所示，`VB1` 在图中出现了两次，但是他们依然应当以 `VB1` 和 `VB$1` 分别命名，作为两个网络。



然而GND和VDD是两个例外。正如参数手册所规定，GND被视作识别的目标。所有以标签或独特标志示意的‘GND’应当被视作同一网络，无论他们在图上是否直接相连。需要注意的是，最终的网表中不应当出现‘GND器件目标’。GND网络的名称不做要求。





同样，VDD也是我们的识别目标。我们的图片中有两种风格的VDD，一种如图（2），VDD用一根长线连接，一种VDD用非连接的短线标识为端口。一张图片里只会包含这两种风格中的一种。对于第二种短线的风格的VDD，虽然在图上他们没有直接相连，但是我们依然视其为同一个网络，并且最终网表里也不应有‘VDD器件目标’。我们确保所有图片的VDD都是同一个电平。注意，只有T字型结构的图标才被认为是VDD，否则我们将其当作普通端口处理，如图（3）。

祝大家参赛愉快！加油！！

更新日期：2024年10月23日

针对GED误差较大的问题进行了处理。由于目前采用的GED算法的时间复杂度较高，得到准确值需要较长的时间，在时间有限的情况下只能牺牲一定的准确性。本次修改后，准确性有所提高，能确保GED为0的网表，即正确识别的网表不会被误判为错；并能立即给出一个近似GED值，供大家参考。为得到更准确的GED值，将需要等待15分钟左右的计算时间。

新的测试报告的“网表识别精确度K”一列下现有两个数值 (a,b)，分别表示立即GED值和计算15分钟后GED值。（这两个值很可能相等）

如果您使用的是服务器上的运行环境，请选择位于 `/bin/python3` 路径下的 `python 3.9.19 64-bit` 解释器。如果您是自行安装的环境，需要新安装支持并行的python包：`pip install distributed` 和 `pip install dask`

更新日期： 2024年10月26日