

2023（第五届）集成电路 EDA 设计精英挑战赛

赛题指南

一、 赛题名称

基于机器学习的 SoC 电源网络静态压降预测

二、 命题企业

杭州行芯科技有限公司

三、 赛题 Chair

林亦波 北京大学

四、 赛题背景

随着大规模集成电路的应用，工艺节点不断缩小，金属互连线宽度变窄，单位长度的电阻越来越大，同时由于芯片的集成度不断提高，导致芯片上远离供电电源的单元电压下降明显，严重的将直接导致该单元的功能出错或失效，这对于设计是致命的，因此 signoff 阶段对电压降的分析至关重要。

电压降主要分为两种。一种是静态的电压降，另外一种则是动态的电压降。

静态电压降现象主要是由于金属连线的自身电阻分压造成的，电流从 PAD 经过电源网络的时候会产生电压降，所以静态电压降主要跟电源网络结构有关，主要考虑电阻效应。

动态电压降是电源在电路开关切换的时候电流波动引起的电压降。这种现象产生在时钟的触发沿，时钟沿跳变不仅带来自身的大量晶体管开关，同时带来组合逻辑电路的跳变，往往短时间内在整个芯片上产生很大的电流，这个瞬间的大电流引起了电压降现象。同时开关的晶体管数量越多，越容易触发动态电压降现象。

对于全芯片的电压降分析，商业软件一般将物理设计抽象为数学模型，然后对大型线性稀疏方程组求解，在上千亿节点的矩阵规模上求解通常需要数百小时的计算，这往往是数字后端设计中最消耗资源的仿真环节。因此我们聚焦 AI for EDA，设计基于‘机器学习的 SoC 电源网络静态压降预测’作为本次挑战赛的题目，通过机器学习方法针对数据集进行标记，用神经网络拟合一个模型进行预测，经过多次和 golden 电压降值进行修正迭代，将预测出来的静态无矢量电压降精度控制在用户可接受的范围内。

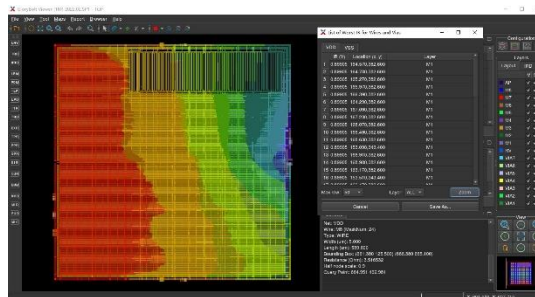


图 1 先进工艺物理设计案例全局平面图

五、 赛题综述

计算电压降首先需要输入 lef 文件、def 文件、itf 文件进行网表抽取，计算电源网络上节点之间的电阻。然后输入 lib 文件、timing 文件、spef 文件，计算出 design 中所有 instance 的 leakage, switching, internal power, 再结合网表和 instance 的功耗文件进行矩阵求解，得到电源网络上的电压和电流信息。现给定每个 design 的功耗值，使用机器学习的方法提取模型并进行静态无矢量压降预测。

下图以物理设计到静态压降估算流程为例进行说明，首先通过寄生参数提取工具得到各节点之间的电阻值，然后通过读取其他 design 文件计算 instance 的功耗值。最后通过 sign off 工具得到精准的静态压降数据形成 golden data。参赛者可通过网表、电阻、功耗等参数提取数据进行标记，建立模型进行预测，通过与提供的 golden data 进行对比修正，优化模型。

本赛题会提供新手代码样例作为参考。

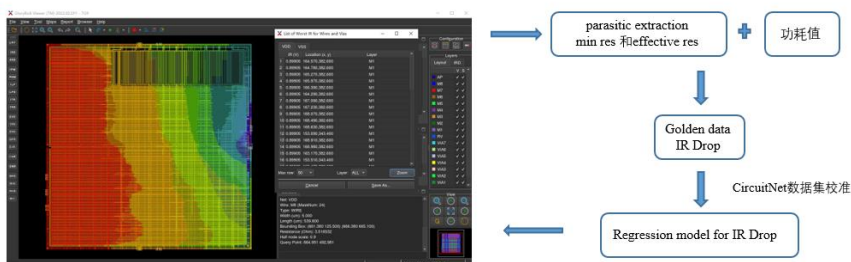


图 2 基于机器学习的静态压降预测模型

- (一) 输入：抽取网表后的 min res path 与 effective res 数据、功耗数据和静态压降数据 (Golden IR Drop)，输入数据由行芯 EMIR 分析工具 GloryBolt 产生。
- min res path 文件中包含 min res value (最小电阻值)、net name (绕线名称)、instance name (单元实例名称)；
 - Effective res 文件中包含：loop_r (instance 连接到 power net 和 ground net 等效电阻值总和)、vdd_r (instance 连接到 power net 的等效电阻值)、gnd_r (instance 连接到 ground net 的等效电阻值)、vdd(x y layer net pin) (instance 在 power net 上的坐标, layer name, net name, pin name 信息)、vss(x y layer net pin) (instance ground net 上的坐标, layer name, net name, pin name 信息)、instance name (单元实例名称)；
 - 功耗数据包含：instance name (单元实例名称)、cell name (instance 所属的 cell 类型名称)、freq(instance 的工作频率)、toggle rate (instance 单位周期的翻转率)、leakage power (instance 的泄露功耗)、switching power (instance 的翻转功耗)、internal power (instance 的内部功耗)、total power (instance 的总功耗)、library (lib 文件名)、P/G-pin (power

pin 名称)、P/G-volt (power net 的理想电压值)、power domain (instance 连接的 power net)、x_y_location(instance bounding box 坐标);

- 静态压降数据包含: inst_vdd (instance power pin 上的实际电压)、vdd_drop (instance power pin 上产生的压降)、gnd_bounce (instance ground pin 上产生的地弹)、ideal_vdd (instance power pin 上的理想电压)、pwr_net (instance 连接的 power net 名字)、location (instance 坐标)、instance name (单元实例名称)。

(二) 输出: 每个 case 的静态压降预测数据

(三) 方法: 机器学习模型, Regression model 为参赛选手自定义的抽象模型参数及相关数值。

(四) 输入数据文件格式:

i. min path res 文件:

```
#MIN RES VALUE    #NET NAME    #INST NAME
```

ii. eff_res 文件:

```
#loop_r  #vdd_r  #gnd_r  #vdd(x y layer net pin)  
#vss(x y layer net pin)  #inst
```

iii. 功耗文件:

```
#instance name #cell name #freq #toggle rate  
#leakage power #switching power #internal power  
#total power #library #P/G-pin #P/G-volt  
#powerDomain #x_y_location
```

iv. Golden data (静态压降文件):

```
#inst_vdd #vdd_drop #gnd_bounce #ideal_vdd  
#pwr_net #location #instance name
```

(五) 输出数据文件格式:

```
#inst_vdd #vdd_drop #gnd_bounce #ideal_vdd  
#pwr_net #location #instance name
```

(六) 输入输出流程图

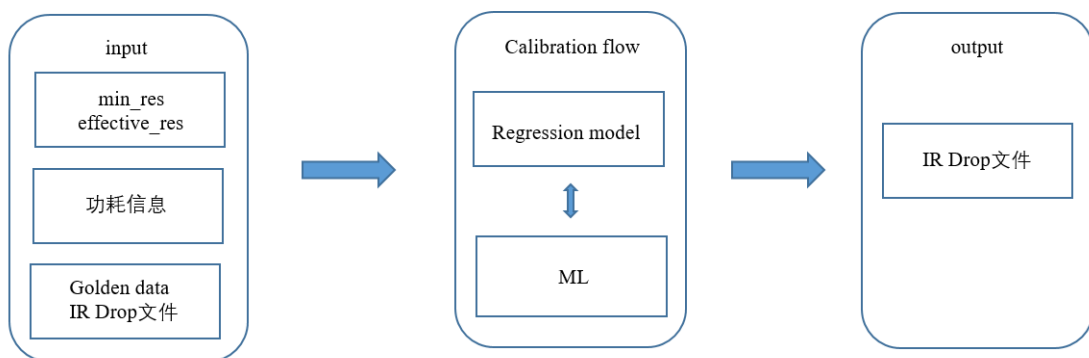


图 3 输入输出流程图

六、 关注指标

对每一个 instance，将 IR Drop 与提供的 golden 数据进行对比，针对给出的 case 对比 instance 上的静态压降的结果精度和运行时间（遇上相同的 instance name 再去匹配 instance 坐标）。行芯公司提供对比脚本。

精度指标关注相关系数与平均绝对误差，性能指标关注运行时间，更高的相关系数、更小的误差和运行时间表示模型更优。

相关系数 CC（0~1 之间），如下定义， y_i 是压降预测值， \hat{y}_i 是 golden 压降值，N 是数据点个数。

$$CC = \frac{\sum_{i=1}^N [y_i - \text{mean}(y)] [\hat{y}_i - \text{mean}(\hat{y})]}{\sqrt{\sum_{i=1}^N [y_i - \text{mean}(y)]^2 \sum_{i=1}^N [\hat{y}_i - \text{mean}(\hat{y})]^2}}$$

平均绝对误差 MAE 计算公式为：

$$\frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$$

具体计算规则为：

若比赛中各队伍的相关系数、平均绝对误差、运行时间的最大值和最小值分别记为 C_{max} 和 C_{min} 、 M_{max} 和 M_{min} 、 T_{max} 和 T_{min} ，其他队伍相关系数、平均绝对误差、运行时间分别记为 C_{mid} 、 M_{mid} 、 T_{mid} ，总得分则为：

$$\text{Score} = \left(\frac{C_{mid} - C_{min}}{C_{max} - C_{min}} \right) * 40\% + \left(1 - \frac{M_{mid} - M_{min}}{M_{max} - M_{min}} \right) * 40\% + \left(1 - \frac{T_{mid} - T_{min}}{T_{max} - T_{min}} \right) * 20\%$$

Score 越大排名越靠前，IR drop 的预测值且 golden 值小于 1mV 的 instance 不计入统计结果。

七、参考资料

[1] 北京大学开源数据集：CircuitNet

数据集：<https://circuitnet.github.io/>

教程样例：<https://github.com/circuitnet/CircuitNet>

[2] V. A. Chhabria, V. Ahuja, A. Prabhu, N. Patil, P. Jain, and S. S. Sapatnekar, "Thermal and IR Drop Analysis Using Convolutional Encoder-Decoder Networks," Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC), 2021.

[3] Chia-Tung Ho and Andrew B Kahng. "IncPIRD: Fast Learning Based Prediction

of Incremental IR Drop,” in the IEEE/ACM International Conference on Computer-Aided Design (ICCAD). 2019.

[4] Zhiyao Xie, Hai Li, Xiaoqing Xu, Jiang Hu, Yiran Chen, “Fast IR Drop Estimation with Machine Learning : Invited Paper,” in the IEEE/ACM International Conference on Computer-Aided Design (ICCAD).2020.

[5] Zhiyao Xie, Haoxing Ren, Brucek Khailany, Ye Sheng, Santosh Santosh, Jiang Hu, and Yiran Chen, “PowerNet: Transferable Dynamic IR Drop Estimation via Maximum Convolutional Neural Network” in Asia and South Pacific Design Automation Conference (ASP-DAC), 2020.

八、 运行环境

需要配置的运行环境如下：

addict==2.4.0

certifi

charset-normalizer==2.1.1

idna==3.3

imageio==2.21.1

joblib==1.2.0

mmcv==1.6.1

networkx==2.8.6

numpy==1.23.2

opencv-python==4.6.0.66

packaging==21.3

Pillow==9.3.0

psutil==5.9.1

pyparsing==3.0.9

pyutil==3.3.0

PyWavelets==1.3.0

PyYAML==6.0

requests==2.28.1

scikit-image==0.19.3

scikit-learn==1.1.2

scipy==1.9.0

threadpoolctl==3.1.0

tifffile==2022.8.12

tqdm==4.64.0

typing_extensions==4.3.0

urllib3==1.26.12

yapf==0.32.0

pytorch == 根据本地 gpu 版本确定

