

# 第四届（2022）集成电路 EDA 设计精英挑战赛

## 赛题指南

一、 **赛题名称**：感知物理信息的智能化时序估算模型

二、 **命题企业**：杭州行芯科技有限公司

三、 **赛题 Chair**：闫浩 东南大学、贺旭 湖南大学

四、 **赛题背景**

传统设计流程中路径时延(path delay)计算以精确模型进行求解，其效率随着电路规模的增大越发受到限制。而先进工艺中线时延 (wire delay) 在时序中所占比重越来越高，而且造成设计过程中迭代周期长，次数多。因此使用神经网络、卷积神经网络等机器学习方法预估延时成为了近些年工业界和学术界的热点，如何在提高效率的同时控制精度误差成为关键。

传统的互连线时序计算方法有线载模型 wire-load models, elmore delay, 渐进波形估计 (AWE), Arnoldi 等，而 path delay 中的主要包含两部分，即 cell delay, wire delay。其中 cell delay 主要基于 dot lib 时序库进行插值查找，精度误差相对可控，而 wire delay 由于绕线的长度不同、密度不同、分布不同、等效模型不同、传播方法不同等因素对延时精度准确计算造成极大影响。对于布局布线阶段精度没有签核 (signoff) 精度要求那么苛刻的场景下可以通过机器学习方法针对大量数据集进行训练、优化，将精度控制在用户可接受的范围内，及早发现时序违例，并修改物理设计，通过布局布线进行多次迭代，最终缩短收敛时间。下图展示了物理设计

阶段的平面图:

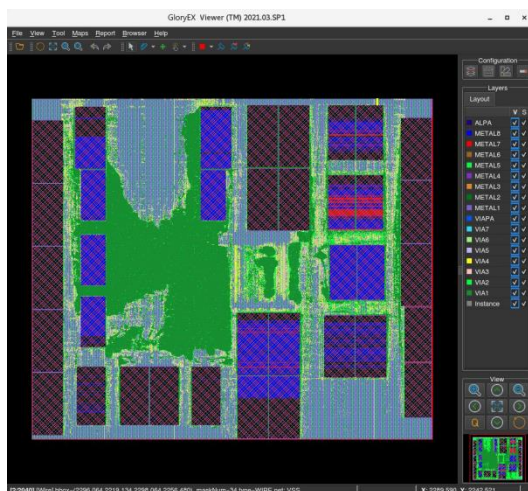


图1 先进工艺物理设计案例全局平面图

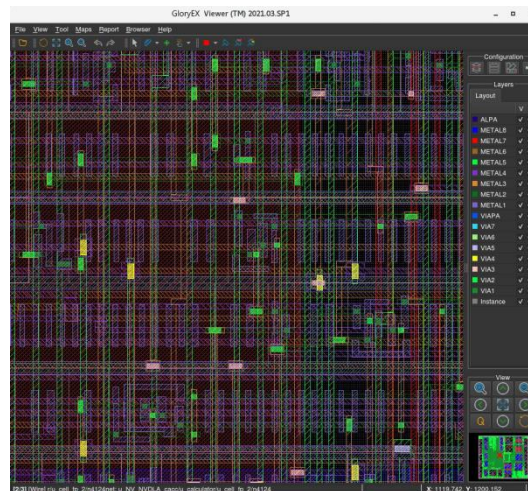


图2 先进工艺物理设计案例局部平面图

## 五、 赛题综述:

通过寄生参数提取技术生成真实物理设计的 SPEF 网表, 通过机器学习方法进行绕线延时预测。现给定设计电路的 SPEF 网表, 包括 net name(绕线名称)、node name(网络节点的名称)、coupling cap(节点耦合电容数值)、net cap(绕线总电容数值)、res(节点电阻阻值), wire delay (绕线延时), 请使用神经网络、卷积神经网络等方法提取模型并进行优化预测绕线延时。

下图以物理设计到延时估算流程为例进行说明, 左图为实际物理设计案例, 首先通过寄生参数提取工具生成SPEF网表, 然后通过spice工具得到精准的绕线延时 (wire delay) 行成 golden data。参赛者可通过节点数量、节点耦合电容、总电容、节点电阻阻值等参数提取模型进行训练, 进而优化模型, 将精度与效率进行很好的平衡, 为芯片设计人员提供高效、高精度的延时计算工具。

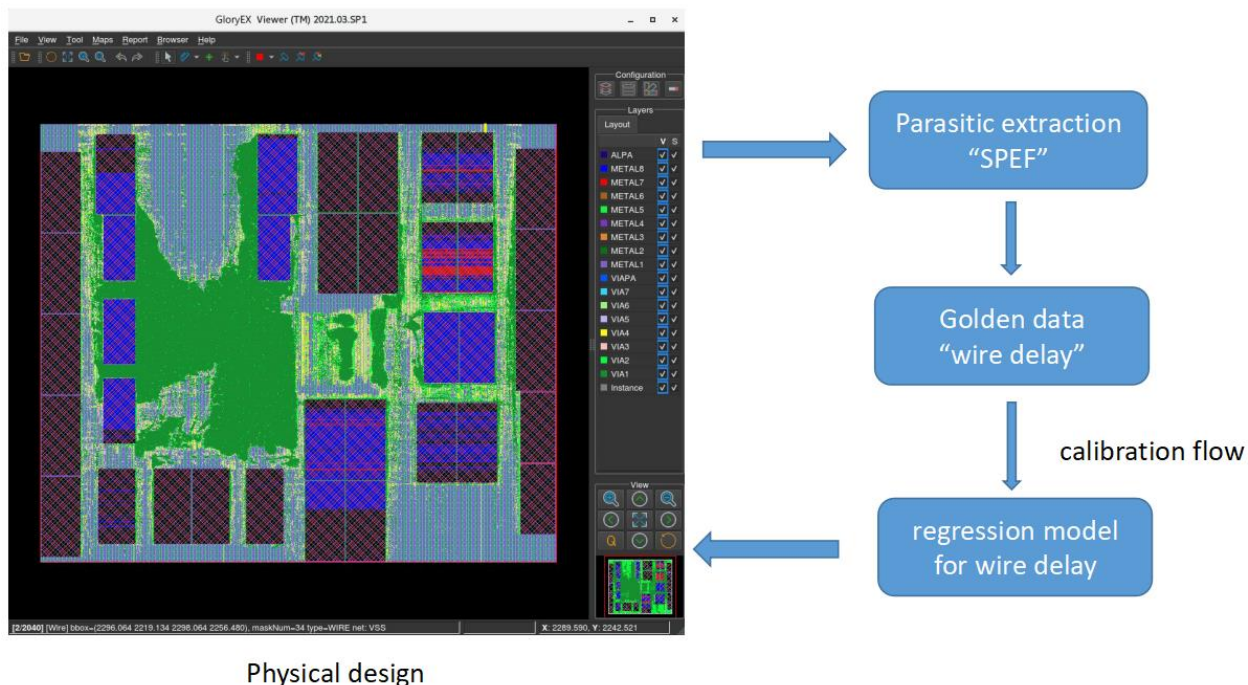


图 1 基于机器学习的时序预测流程

1) 输入：物理设计之后生成的 SPEF 网表 (RC netlist)、Netlist 相关信息文件和延时数据 (wire delay)。

其中 RC 网络进行了切分，该数据包括 net name(绕线名称)、node name(网络节点的名称)、coupling cap(耦合电容数值)、net cap(绕线总电容数值)、res(电阻阻值)；

netlist 相关信息文件包含 net name(绕线名称)、output pin (输出端口)、input pin (输入端口)、pin cap (端口电容)；

延时数据中包含 net name (绕线名称) 和 wire delay (绕线延时)；

Regression model 为参赛选手自定义的抽象模型参数及相关数值。

2) 输入要求：RC 网络没有电阻回路，所有电容都是节点对地电容。Regression model 相关参数及数字定义说明要清晰，可适用于任意绕线的延时计算。

3) 输出：每条连线输出端口名称和输入端口名称 (output pin name, input pin name)，连线总延时 (wire delay)，运行总时间 (runtime)。

4) 方法: 机器学习模型。

5) 提交要求:

源工程项目代码, Linux 执行码, 设计文档, 包括架构, 算法, 测试报告

执行码的命令行和参数要求如下:

```
WireDelayCalc -in xxx -out xxx.out
```

6) 输入数据文件格式

1. RC netlist 文件

```
(C_UNIT 1PF, R_UNIT 1OHM)
```

```
*D_NET *0001 0.0001
```

```
*RES
```

```
1 node1 node2 res_value
```

```
2 node2 node3 res_value
```

```
*CAP
```

```
1 node1 cap_value
```

```
2 node2 cap_value
```

2. Netlist info 文件

```
Net name: cts1
```

```
Output: cts1/ZN //output pin name
```

```
Input: cts1/I //input pin name
```

```
Pin Cap. : 0.001 //input pin cap value
```

```
Input: cts1/2
```

```
Pin Cap : 0.002
```

Input: cts1/3

Pin Cap : 0.003

3. Golden data: Wire delay 文件

OutputPin InputPin WireDelay(ns)

cts1/ZN cts1/l 0.05

7) 输出文件格式

OutputPin InputPin WireDelay(ns)

cts1/ZN cts1/l 0.01

8) 输入输出流程图

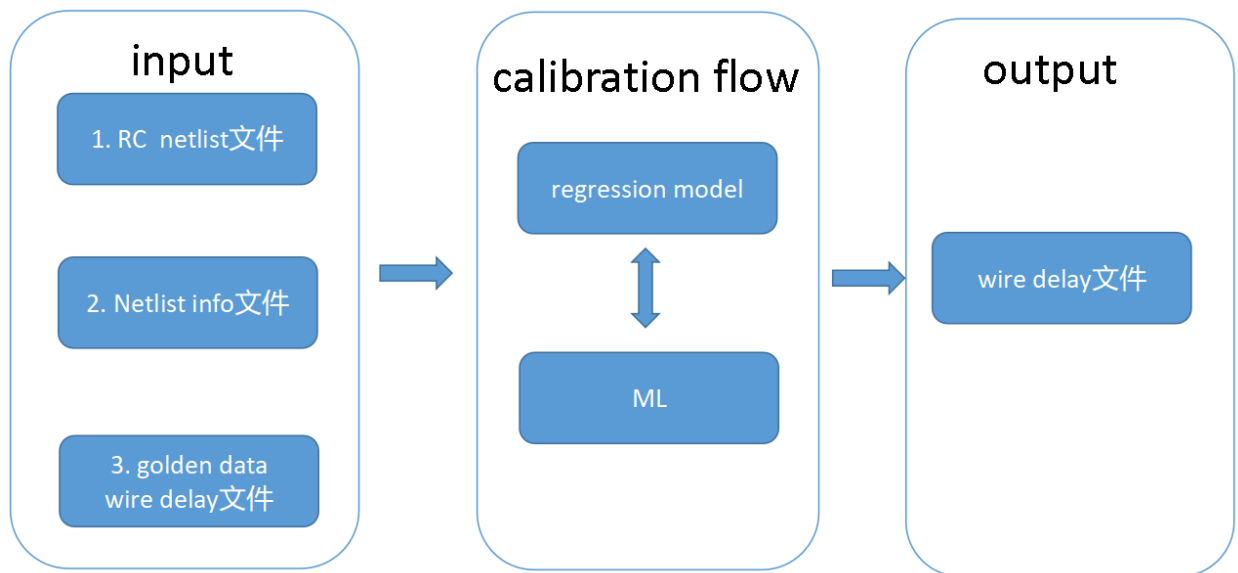


图 4 输入输出流程图

**关注指标:**

1.对每一条绕线, 将带有物理信息的 wire delay 与提供的 golden 数据进行对比, <50ps 时按绝对误差评估精度, >50ps 时按相对误差评估精度, 误差定义如下:

a) 大于 50ps 则要求相对误差:

$\text{mean} < 5\%$ ,  $2\text{sigma} < 10\%$ ,  $\text{max} < 20\%$ ;

其中随着 wire 长度越大, C 分布的复杂程度, Node 数量越多, fanout 数值越大, 均对延时精度的影响越大, Net cap 数值越大的要求相对误差越小。

b) 小于 50ps 则要求绝对误差:

$\text{mean} < 10\text{ps}$ ,  $2\text{sigma} < 20\text{ps}$ ,  $\text{max} < 50\text{ps}$ ;

其中随着 wire 长度越小, C 分布的复杂程度, Node 数量的多少对延时精度的影响越大, fanout 数值越大, Net cap 数值越小的要求绝对偏差越小。

2. 总体运行时间越小越好, 低于参考时间后按照大小进行排名。

## 测试例子

1. 基于先进工艺案例, 提供 100 组数据, 每组数据 wire delay 中可能同时包含大于 50ps 和小于 50ps 的, 每组一万条 NET 的样本;

2. 基于不同于训练使用 RC Corner 的测试数据 10 组, 5 组大于 50ps 的 NET, 5 组小于 50ps 的 NET, 每组一万条 NET。

## 评分标准

总分 100, 测试 10 组数据, 每组 10 分。在每组内评分中, 精度占 8 分, 运行时间 Run Time 占 2 分。

对于每组数据, 同时满足准确度和 Run time 要求得满分, 5 组大于 50ps, 5 组小于 50ps:

a) 精度: 每组 8 分, 共 10 组。

大于 50ps 的五组:

- $\text{max} < 15\%$ , 为 A 档, 得 2~4 分, 按参赛队伍名次折算;  $15\% < \text{max} < 30\%$ , 为 B 档, 得 0~2 分, 按参赛队伍名次折算;  $30\% < \text{max}$  为 C 档, 得 0 分。

- $2\sigma < 8\%$ , 为 A 档, 得 1~2 分, 按参赛队伍名次折算;  $8\% < 2\sigma < 15\%$ , 得 0~1 分, 按参赛队伍名次折算;  $15\% < 2\sigma$ , 为 C 档, 得 0 分;
- $\text{mean} < 3\%$ , 为 A 档, 得 1~2 分, 按参赛队伍名次折算;  $3\% < \text{mean} < 5\%$ , 为 B 档, 得 0~1 分, 按参赛队伍名次折算;  $5\% < \text{mean}$ , 为 C 档, 得 0 分;

小于 50ps 的五组:

- $\text{max} < 20\text{ps}$ , 为 A 档, 得 2~4 分, 按参赛队伍名次折算;  $20\text{ps} < \text{max} < 30\text{ps}$ , 为 B 档, 得 0~2 分, 按参赛队伍名次折算;  $30\text{ps} < \text{max}$ , 为 C 档, 得 0 分。
- $2\sigma < 20\text{ps}$ , 为 A 档, 得 1~2 分, 按参赛队伍名次折算;  $20\text{ps} < 2\sigma < 30\text{ps}$ , 为 B 档, 得分 0~1 分, 按参赛队伍名次折算;  $30\text{ps} < 2\sigma$ , 为 C 档, 得 0 分;
- $\text{mean} < 10\text{ps}$ , 为 A 档, 得 1~2 分, 按参赛队伍名次折算;  $10\text{ps} < \text{mean} < 15\text{ps}$ , 为 B 档, 得 0~1 分, 按参赛队伍名次折算;  $15\text{ps} < \text{mean}$ , 为 C 档, 得 0 分;

b) b) 性能: 每组 2 分, 共 10 组。

每组运行总时间, 相对于参考时间:

- 性能小于参考时间, 为 A 档, 得 1~2 分, 按参赛队伍名次折算;
- 大于参考时间 50%~100%, 为 B 档, 得 0~1 分, 按参赛队伍名次折算;
- 大于参考时间 100%, 为 C 档, 得 0 分;

## 参考资料

[1]Seung-Soo Han, Andrew B Kahng, Siddhartha Nath, and Ashok S Vydyanathan. 2014. A deep learning methodology to proliferate golden signoff timing. In Proc. DATE. 1-6

[2]ACM TAU 2015 Contest: Incremental Timing and Incremental CPPR", 2015, [online] Available: <https://sites.google.com/site/tacontest2015/home>

[3]Andrew B Kahng, Seokhyeong Kang, Hyein Lee, Siddhartha Nath, and Jyoti Wadhvani. 2013. Learning-based approximation of interconnect delay and slew in signoff timing tools. In Proc. SLIP. 1–8.

[4]Andrew B Kahng, Mulong Luo, and Siddhartha Nath. 2015. SI for free: machine learning of interconnect coupling delay and transition effects. In Proc. SLIP. 1–8.

[5]Fathi, Bahareh, Laleh Behjat, and Logan M. Rakai. "A pre-placement net length estimation technique for mixed-size circuits." Proceedings of the 11th international workshop on System level interconnect prediction. 2009.

[6]A. B. Kahng and S. Reda, "Intrinsic shortest path length: a new, accurate a priori wirelength estimator," in ICCAD, 2005.

[7]Q. Liu and M. Marek-Sadowska, "Pre-layout wire length and congestion estimation," in DAC, 2004.

[8]GitHub 资源: <https://github.com/tensorflow/tensorflow>