敏捷设计中基于机器学习的静态时序分析方法综述

贺旭[1)], 王耀[2)], 傅智勇[1)], 李暾[2)], 屈婉霞[2)*], 万海[3)], 张吉良[1)]

[1)] (湖南大学信息科学与工程学院 长沙 410082)
[2)] (国防科技大学计算机学院 长沙 410073)
[3)] (清华大学软件学院 北京 100084)
(dawn.hx@gmail.com)

摘 要: 随着集成电路规模越来越大, 设计变得越来越复杂. 为了有效地提升设计生产率, 芯片敏捷设计受到越来越广泛的重视. 在芯片 RTL-to-GDSII 设计流程中, 敏捷设计方法需要广泛借助机器学习技术, 寻求“无人参与”的解决方案. 时序性能作为芯片的重要性能指标, 需要在 RTL-to-GDSII 设计的各个流程中进行静态时序分析. 快速、准确、可靠的时序预测, 可以将 Sign-Off 的时序性能前馈到早期设计流程中, 指导早期设计的时序优化和时序收敛, 减少芯片设计的迭代次数和迭代周期. 文中给出敏捷设计中时序优化的流程框架, 详细地梳理了 RTL-to-GDSII 设计流程中基于机器学习的时序分析研究现状; 并从数据准备、问题建模、实用性以及通用性等多方面, 探讨了敏捷设计中基于机器学习方法进行时序预测的挑战.

关键词: 敏捷设计; 电子设计自动化; 静态时序分析; 机器学习
中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2023.19557

A Survey on Machine Learning-Based Technology for Static Timing Analysis in Agile Design

He Xu[1)], Wang Yao[2)], Fu Zhiyong[1)], Li Tun[2)], Qu Wanxia[2)*], Wan Hai[3)], and Zhang Jiliang[1)]

[1)] (*College of Computer Science and Electronic Engineering, Hunan University, Changsha* 410082)
[2)] (*College of Computer Science and Technology, National University of Defense Technology, Changsha* 410073)
[3)] (*School of Software, Tsinghua University, Beijing* 100084)

Abstract: As integrated circuits (ICs) become larger and more complex than ever, to increase design automaticity and productivity, agile design methodology has attracted a lot of attentions. In back-end design of ICs, machine learning technology for agile design are required to build a no-human-in-the-loop RTL-to-GDSII flow. For chip design, timing performance is a critical but effort-taking task. An accurate timing predictor, which is highly correlated with Sign-Off timing, is desirable to guide the timing optimization in the early design process. In this work, we propose a feasible framework for timing optimization in agile design. We also give discussion of the prior researches of machine learning-based timing predictions in RTL to GDSII design process in detail. At last, we further summarize the challenges of timing prediction in agile design from the aspects of data preparation, problem modeling, practicality and generality.

收稿日期: 2022-01-02; 修回日期: 2022-07-23. **基金项目**: 国家自然科学基金(61872136, U19A2062). **贺旭**(1985—), 女, 博士, 教授, 博士生导师, CCF 会员, 主要研究方向为物理设计、可制造性设计、静态时序分析; **王耀**(1983—), 男, 博士, 助理研究员, 主要研究方向为嵌入式芯片设计、类脑计算; **傅智勇**(1997—), 男, 硕士, 主要研究方向为静态时序分析; **李暾**(1974—), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究方向为电子设计自动化; **屈婉霞**(1972—), 女, 博士, 副研究员, 论文通信作者, 主要研究方向为体系结构、处理器设计; **万海**(1981—), 男, 博士, 副研究员, 硕士生导师, 主要研究方向为实时以太网、形式化建模与验证; **张吉良**(1986—), 男, 博士, 教授, 博士生导师, 主要研究方向为硬件安全.

Key words: agile design; electronic design automation; static timing analysis; machine learning

1　敏捷设计方法

随着集成电路(integrated circuit, IC)制造工艺的特征尺寸不断缩小，芯片的复杂度和芯片设计成本不断增加. 2001 年，国际半导体技术路线图(international technology roadmap for semiconductors, ITRS)[1]曾指出IC设计成本是制约半导体发展的最大威胁. 随着 IC 行业分工不断深化，在芯片设计、电子设计自动化(electronic design automation, EDA)工具、芯片制造等各产业链环节之间，由于缺乏对全流程专业知识的整合能力，也进一步制约了先进工艺节点下 IC 产品的开发.

解决上述问题可以有效地降低芯片设计成本，缩短设计周期，提高设计生产率. 近年来，研究人员借鉴软件设计中面向对象编程的思想，在工业界和学术界将敏捷设计理念引入芯片设计流程. 不同于现有芯片设计方法以频率、功耗、面积等性能指标为导向的紧耦合设计方法，敏捷设计采用松耦合设计方法，以时间、成本、复杂度等敏捷度指标为导向并兼顾性能. 在敏捷设计中，对整个芯片系统采用面向对象体系结构(object-oriented architecture, OOA)进行解耦，实现高于门级、但低于或等于基础模块级的硬件对象及封装，为芯片上层体系结构设计提供清晰、直观的抽象模型和开发框架，实现一种灵活的结构定制方法，从而使芯片微体系结构的设计空间探索更加高效且易于维护.

现有的芯片敏捷设计方法在系统上层设计中采用 OOA，通过提升现有硬件设计描述语言的抽象层次，建立连接高层语言和底层硬件的中间表示形式，以方便硬件设计的复用和优化；同时，通过一系列硬件综合技术和辅助代码调试工具将上层体系结构参数传递到后续 RTL-to-GDSII 设计自动化工具. 其中，寄存器传输级(register-transfer level, RTL)代码指体系结构设计后进入逻辑综合的设计输入文件(即高层设计产生的硬件描述文件)，表示物理实现的起点；GDSII 则指版图输出文件，表示物理实现的终点. 在 RTL-to-GDSII 流程中，敏捷设计方法通过对高层体系结构特征参数空间进行高效的物理设计方案搜索寻优，实现在 RTL-to-GDSII 流程中融合高层体系结构信息，并支持高层体系结构设计与物理设计协同优化，以快速完成满足设计要求的物理实现[2].

芯片设计从 RTL 到 GDSII 的流程跨越了逻辑综合、布局规划、布局、时钟树综合(clock-tree synthesis, CTS)、布线、寄生参数提取(resistance and capacitance, RC extraction)、静态时序分析(static timing analysis, STA)和签核(Sign-Off)检查等多个阶段. 为提升电路设计自动化程度，美国国防高级研究计划局(defense advanced research projects agency, DARPA)已经发布的 IDEA(intelligent design of electronic assets)项目[3]中就包括寻求“无人参与”的 RTL-to-GDSII 设计流程.

目前，芯片设计质量在很大程度上取决于芯片设计师的经验，智能 EDA 工具通过学习已有的设计经验可以加速设计流程. 已有的机器学习(machine learning, ML)技术涉及 EDA 领域的多个应用场景，如考虑异常检测的成品率模型[4]、光刻热点检测[5]、数据路径的规则识别[6]、验证测试集生成[7]、模拟电路性能建模[8]以及设计空间探索[9]等. 通过 ML 方法对电路的设计过程进行关键参数提取、可预测性建模以及 EDA 算法优化，可有效地提升芯片 RTL-to-GDSII 设计效率，缩短芯片版图的生成时间，在一定程度上达到敏捷设计的目标.

本文针对敏捷设计 RTL-to-GDSII STA 问题进行讨论. 首先简要介绍 STA 及其在敏捷设计下的问题背景和应用框架；然后对当前国内外 ML STA 相关工作进行综述；最后总结敏捷设计中 STA 未来所面临的挑战.

2　敏捷设计中的 STA

芯片敏捷设计方法需要解决的一个首要问题是 EDA 工具的运行时间[10]. 为了确保 Sign-Off 结果满足设计的时序约束要求，在 RTL-to-GDSII 流程中需要多次调用 STA 工具，并有可能需要经过多次设计迭代才能达到预期目标. 每次 STA 分析的运行时间可能长达数小时；对于较大的设计，该过程甚至需要数周或数月的时间. 此外，布局和布线(placement and routing, PR)工具的中间结果时序报告和 Sign-Off 时序报告结果可能会存在一致性问题，将会导致 PR 过程中时序优化不够精准，影响功耗-速度-面积等性能优化[11]，甚至导致额外的设计迭代周期. 为了提高设计效率，减少设计迭代次数，需要在设计流程中引入更为有效的时序分析方法，以提高 PR 过程中各个中间阶段时序优化

的精准度，加速时序的收敛过程.

为了详细阐述敏捷设计中的 STA 过程，首先给出 STA 的问题描述，并介绍其应用场景；然后讨论敏捷设计中 STA 的需求，并给出基于 Sign-Off 前馈的“预测-优化”框架.

2.1 STA 问题描述

对于给定的芯片设计、目标时钟频率以及外部环境条件等因素，STA 的目标是验证该设计在给定的外部环境和目标频率条件下是否存在时序违反. STA 分析通常包括建立时间检查和保持时间检查 2 种类型. 其中，建立时间检查用于确保数据信号在给定时钟周期内能按时抵达寄存器；保持时间检查用于确保数据信号在被发起采样后能在一定时间内保持有效，以保证寄存器不会漏采信号.

图 1 所示为电路网表转化为对应的 DAG 的过程. 在 STA 中，电路网表会被建模为有向无环图(directed acyclic graph, DAG)，如图 1b 所示. 电路图中的输入/输出端口(primary input & output port, PIO)和引脚对应 DAG 的节点；线网连线或门内部的引脚间时序弧对应 DAG 的边，时序弧的时延对应边的权重.

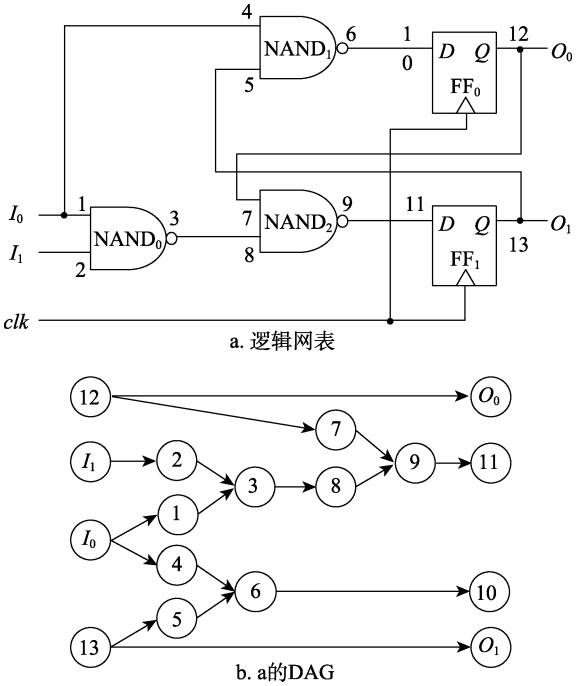


图 1 时序图举例

在 DAG 中可对所有的节点进行拓扑排序，并按层遍历计算出每个节点的到达时间(arrival time, AT)；然后根据终端节点 endpoint，包括寄存器数据端和主要输出端口的时序要求，反向计算出电路图中每个节点的要求到达时间(require arrival time, RAT)；最后根据节点的 AT 和 RAT 的差异得到每个节点的时间裕量(slack)，若裕量为负，则该路径不满足时序要求，需要在后续设计中对其进行时序优化.

根据 DAG 中的时序弧属于门电路内部时延还是属于门与门之间的连线时延，可以分为门时延和线时延. 在 STA 中，门时延可以根据工艺库 Lib 文件中元器件特性，通过输出负载和输入引脚的信号跳变时间(slew)值用查表法计算得到. 线时延则可根据布线信息用线时延模型求出.

2.2 STA 的应用场景概述

时序收敛是电路重要性能指标. STA 会被应用在 RTL-to-GDSII 流程的各个阶段，如图 2 所示，每个设计阶段必须保证 STA 的结果在当前阶段满足时序约束，才能进入下一阶段. 如果当前阶段出现时序违反且无法通过当前阶段的时序优化解决，则需要返回上一级重新设计，从而导致设计迭代.

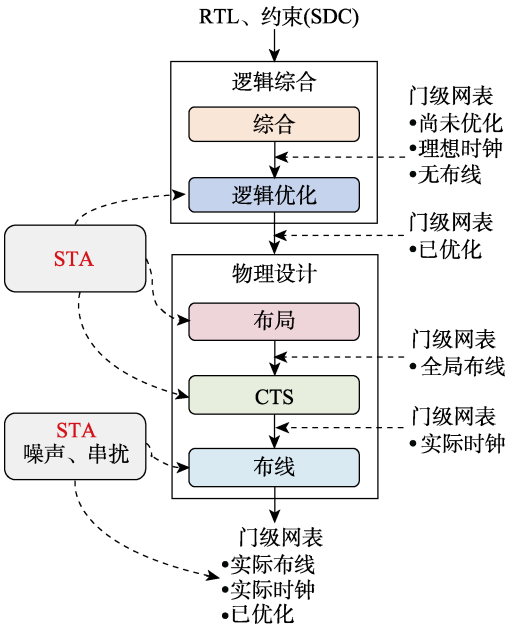


图 2 STA 应用在 RTL-to-GDSII 流程各个阶段

由此可见，STA 是检测各个设计阶段中间结果是否满足时序收敛的重要手段，各个阶段的 STA 准确性直接影响整个设计流程.

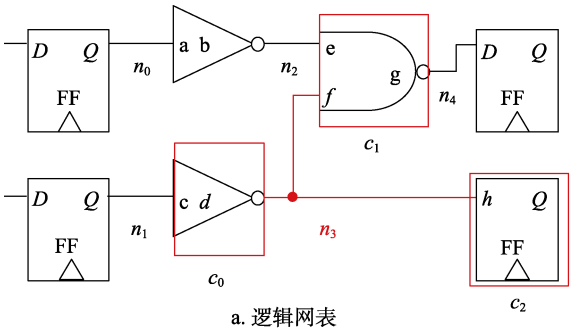
2.3 敏捷设计中的 STA 问题

在敏捷设计方法中，电路设计流程中无人参与的前提是对流程中各个阶段的电路性能都可以进行自动、准确的分析和优化.

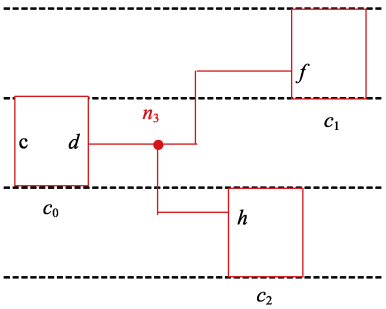
在时序性能分析中，判断 STA 的准确标准是各个阶段的 STA 结果与最终 Sign-Off 阶段的时序报告结果的一致性程度.

然而，由于 RTL-to-GDSII 流程的中间设计阶段提供的电路信息有限，往往造成中间阶段的时序分析结果与最终 Sign-Off 的时序结果不一致.

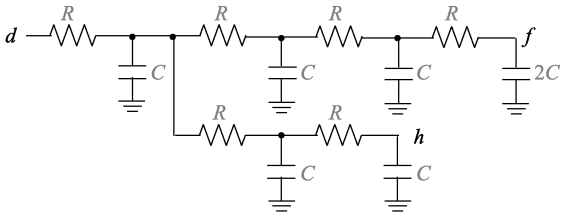
图 3 所示为同一设计在不同阶段的电路信息示例. 逻辑综合阶段将产生电路的逻辑网表，在布局布线阶段分别得到电路的坐标信息和走线信息；根据走线信息可以产生每个线网对应的 RC 图. RC 图是进行准确 STA 的基础.



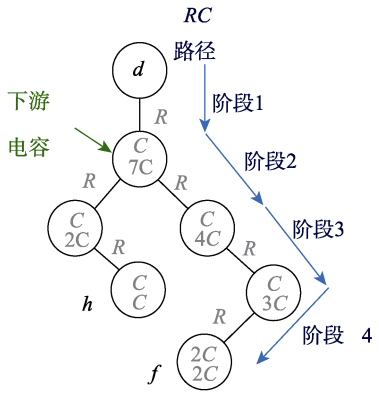
a. 逻辑网表



b. 线网n_3布局布线示例



c. 线网n_3的RC网络



d. 线网n_3的RC图

图 3　RTL-to-GDSII 流程各阶段产生信息示例

由于在 Sign-Off 完成之前无法得到每个线网准确的 RC 信息，因此不同阶段的 STA 会根据 RC 信息的完整度从以下建模方式中选取相适用的方法[12]：(1) 互连线的建模方式. 包括理想互连线、线负载模型、具有近似 RC 值的全局布线以及具有精确 RC 级的实际布线；(2) 时钟的建模方式. 包括零偏差的理想时钟或者实际传播时钟；(3) 信号完整性(signal integration, SI)的建模方式. 如是否考虑信号之间的耦合效应以及分析串扰噪声等.

本文给出敏捷设计下时序的预测-优化总体框架，如图 4 所示. 在敏捷设计中，为了帮助提高 RTL-to-GDSII 流程中时序预测与 Sign-Off 时序分析结果的一致性，该框架采用基于 ML 的时序预测方法，指导 RTL-to-GDSII 流程中各个阶段的时序预测和优化过程，大幅度减少设计迭代次数，甚至达到一次通过的目标；此外，为了使上层体系结构设计方案得到及时反馈，需要对上层体系结构各个硬件对象的特征参数进行 RTL-to-GDSII 各个阶段的设计跟踪，所得数据可以作为评估上层体系结构设计方案的指导依据. 因此，敏捷时序建模解决的主要问题是以 RTL-to-GDSII 流程的数据为依据，包括上层体系结构设计和 RTL-to-GDSII 等各个阶段的可预测性时序评估，消除现有工具流程中时序分析过程存在的过度悲观估计的问题，使得设计评估过程更为准确，从而减少各流程间的设计迭代次数和时序悲观分析等因素，减小设计面积和功耗开销，提高芯片性能，缩短设计的迭代周期.

3　基于 ML 的时序分析现状

目前，绝大多数电路设计采用商用 Sign-Off STA 工具对设计进行时序和关键路径分析，并将其结果作为流片最终依据；极少数需要采用 SPICE 仿真结果作为最终依据. 但是，不论哪种方式，在 Sign-Off 阶段之前都存在时序信息提取不完全问题. 此外，即使处于 Sign-Off 阶段，由于增量式分析需求或者不同时序工具结果不一致性，以及不同模式下时序分析精度和运行时间的权衡问题，都要求改进 STA 分析工具的分析准确度，提高分析效率，从而使得预测结果和 Golden STA 更加一致，即提高时序预测结果的一致性.

表 1 所示为不同场景下基于 ML 的代表性工作. 下面就各场景下具体工作进行说明.

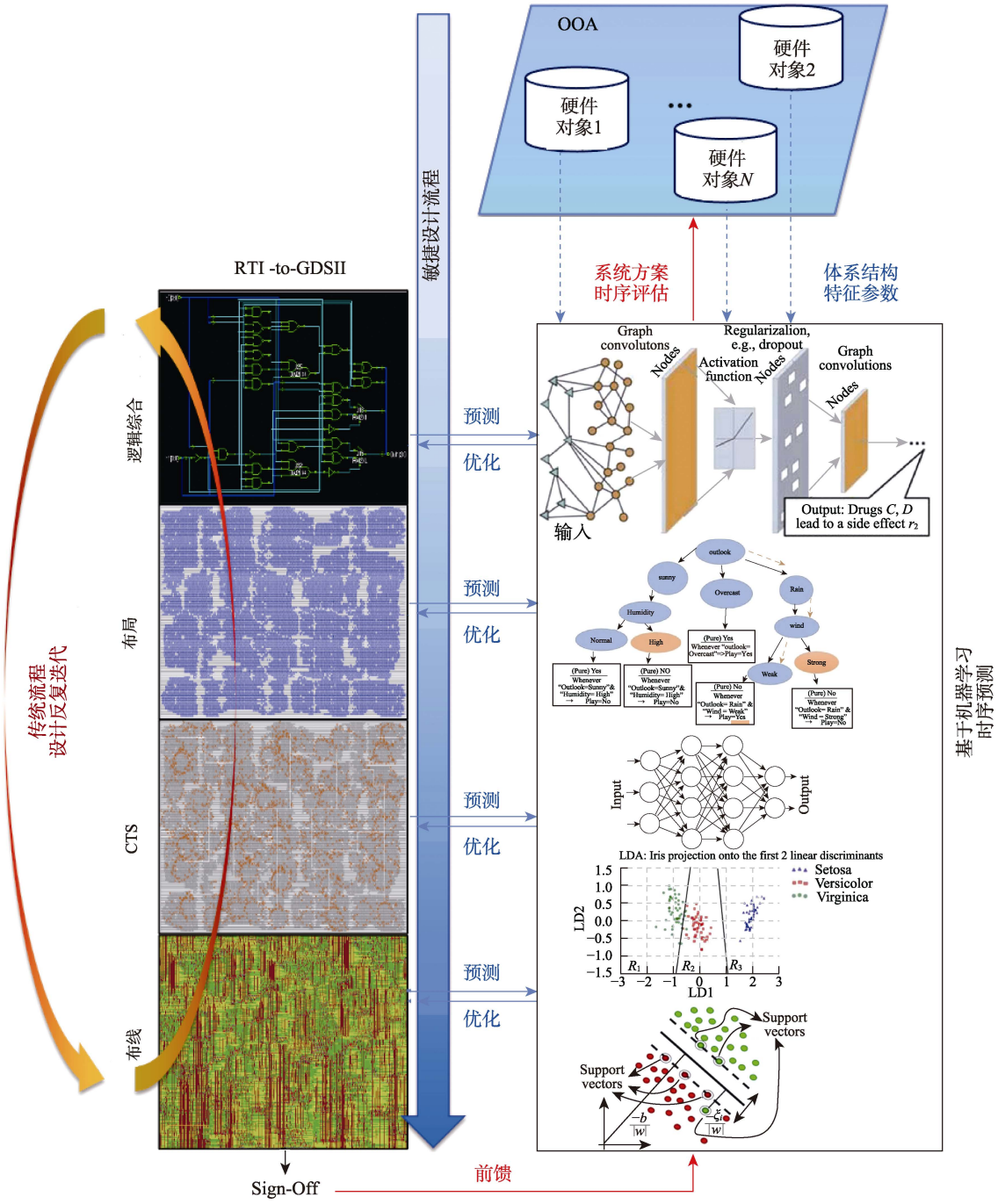


图 4 敏捷设计的时序预测-优化框架示意图

表 1 基于 ML 的时序分析汇总

应用场景	模型目标	文献
逻辑综合	线长预测、关键路径分类	[13-21]
布局阶段	线网时延预测	[22]
CTS	时钟树生成	[23-27]
布线阶段	线时延预测	[28]
Sign-Off 阶段	Sign-Off 工具间预测、基于非 SI 模式预测 SI 模式、基于 GBA 预测 PBA	[29-32]
其他	多工艺角延迟预测、 GPU 加速	[33-41]

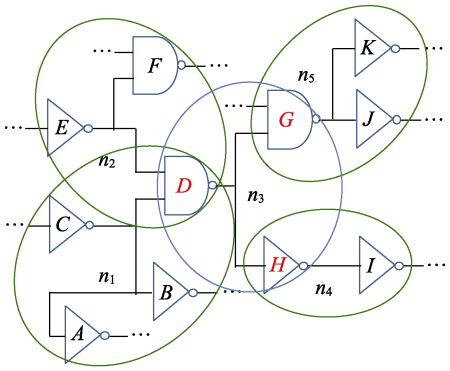
3.1 逻辑综合阶段

逻辑综合阶段以 RTL 设计作为输入，输出综合后的逻辑网表，而在逻辑综合过程中，需要对网表进行 STA 分析以评估设计的时序性能并优化. 由于逻辑综合阶段没有互连线的线长、电容和电阻等信息，设计人员通常先使用线负载模型根据扇出单元的数量估算线网的长度，再根据模型中定义的互连线单位电容电阻值预估互连线的电容和电阻.

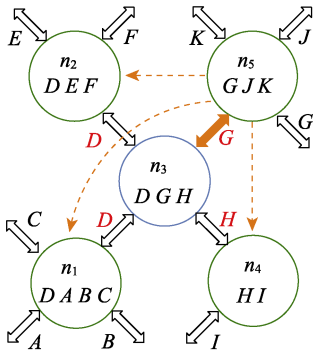
为了更准确地得到互连线信息，需要在逻辑

网表级进行线长预测. 文献[13-14]预先定义了描述每个线网的大量特征，通过拟合的方法为这些特征建立多项式模型，并进行线长预测；文献[15]提出一种人工神经网络(artificial neural networks, ANN)的方式来估计 FPGA 设计中的布线长度；文献[16]通过相互收缩方法(mutual contraction, MC)检查每个相邻线网中的单元数量来估计线长；文献[17]利用内在最短路径长度(intrinsic shortest path length, ISPL)找到待估计线网中单元之间的最短路径，估计线网内单元的亲疏关系，并以此作为判断线网线长依据；与文献[17]中的方法类似，文献[18]利用网表中单元之间的连接关系的向量距离来估计线长；文献[19]则采用基于虚拟 PR 的结果来估计整个路径的线长，但其不适用于单个线网线长预测.

为了以全局的视角进行逻辑网表信息提取来更准确地预测线长，文献[20]提出一种基于图神经网络(graph neural network, GNN)的“Net^2”线长模型. 在 GNN 模型中，网表被转换成一个有向图，每个线网表示为一个节点；线网之间通过公共单元进行扇入和扇出连接，2 个线网间的公共单元称为边单元. 如图 5 所示，通过边单元 G 和 H，n_3 分别连接 n_4 及 n_5；以驱动门 D 为边单元，n_3 与 n_1 及 n_2 相连.



a. 逻辑网表



b. 以线网为节点连接图

图 5　GNN 中以线网为节点的建图示例

在 GNN 模型中，每个线网节点的特征包括其在图中的扇入/扇出相连线网的个数、当前线网的驱动面积、当前线网的扇出门总面积以及其相连线网的各自扇入/扇出个数等信息. 每条边的特征需要考虑边单元对目标线网线长的拉力影响. 为了给拉力定量，对图中线网节点以及原始网表单元分别采用多级划分方法 hMETIS[42]，将线网/单元划分为多个集群. 一般而言，划分到不同集群的单元/线网往往认为其布局后也距离较远，根据划分结果提供全局的视角判断线网/单元之间的亲疏性，得到边单元对目标线网的拉力信息，作为与距离相关的边特征.

文献[20]提出的 GNN 模型本身和特征提取都考虑了线网的局部关系以及一定范围内的全局拓扑关系. 实验结果表明，预测线长有较高的准确度.

值得一提的是，在逻辑综合阶段，无论是预测线长并基于预测线长计算时延，还是其他时序预测目标，如基于网表预判关键路径和非关键路径[21]，都会面临的问题是如何得到样本在 Sign-Off 阶段的答案标签(Ground-Truth). 这是因为同一逻辑综合输出的网表样本可对应到物理设计阶段多种不同实现，这些不同实现来自于芯片面积和单元密度等多种优化参数的不同设置. 因此，逻辑综合中网表所预测的线长或时延，其 Ground-Truth 本身就有很多不确定因素. 从实用性出发，逻辑综合阶段的预测模型应该不能仅局限于单一设计参数设置下的预测准确度，还需要具备不同设计参数设置下的普适性，并围绕这种普适性问题进行预测建模和准确度评估.

3.2　布局阶段

在布局阶段，由于尚未进行布线，因此连线没有具体的电阻电容信息，将导致门时延和线时延无法准确计算. 文献[22]将门时延和线时延合并在一起进行预测，即采用所谓的 net-based delay，其 net-based delay 是指从 driver 到其对应的一个 sink 接收引脚时延，包括 driver 时延及其到 sink 的连线时延. 为了简化模型，该模型不区分 driver 的不同输入引脚以及上升/下降的不同时延，而统一采用最差时延进行预测.

由于在布局阶段没有布线信息，文献[22]中时延的时序特征及其处理方法包括：(1) 线网 driver 和其所有 sink 总电容，driver 输出电容通常与其驱动强度成正比；此外，sink 总电容为 driver 提供负载，这 2 种电容都是线网延迟和 slew 计算的决定性因素. (2) 线网 driver 和目标 sink 之间的距离，

driver 距离目标 sink 的水平和垂直距离通常与相应的线时延成正比. (3) 线网 driver 输入 slew 最大值, slew 定义为信号跳变时间, slew 越大信号跳变速度越慢, 线网延迟和 sink slew 都受 driver 输入 slew 的影响. 由于不同类型的逻辑门的输入引脚数目不同, 而 ML 模型通常需要固定的输入大小, 因此文献[22]使用线网 driver 所有输入引脚之中的最大 slew 值作为特征. (4) 上下文 sink 的位置信息在估计目标 sink 的延迟时, 与其同一线网的其他 sink 充当上下文 sink. 由于不同线网上下文 sink 数目并不相同, 而 ML 模型需要固定大小的输入, 因此, 文献[22]中使用所有上下文 sink 的中值以及各 sink 位置在 X/Y 坐标系中的标准差来表示上下文 sink 的分布情况.

文献[22]选择随机森林方法进行模型训练和预测. 对于给定的测试电路, 先基于训练模型预测测试电路的时序信息, 再采用 PERT 算法按前后层级顺序计算出每个端点的 slack. 若是终端节点, 即寄存器或输出端口的 slack 为负, 则判定为关键路径.

总体而言, 该工作作为布局阶段首篇基于 ML 的时序预测方法, 在 ML 模型上进行了多种尝试. 此外, 对样本采集和时序特征提取提供了指导性思路. 未来可以通过将时序预测模型更精细化来进一步减少时延悲观分析, 时序相关的特征提取也需要完善, 以提升预测准确度.

3.3 CTS

CTS 是物理设计的一个关键阶段. 时钟树可以保证时钟域中寄存器的时钟边沿偏差最小, 从而保证良好的时序特性. 然而, 由于时钟网络的功耗在芯片总功耗中占比较大, 且与布线和时序性能息息相关, 因此优化时钟树有助于避免某些严重的设计问题, 包括过高的功耗、布线拥挤、时序收敛太慢等问题[43]. 设计师为综合出高质量的时钟树, 必须在范围广泛的候选参数中搜索, 找到合适的输入参数以优化关键指标, 如时钟功率、偏斜、时钟线长等. 目前, CTS 调参的过程通常是手动完成, 校准过程十分耗时.

为了减轻设计者的负担, 在 CTS 自动化方面, 文献[23]利用数据挖掘工具来估计 skew 和插入延迟; 文献[24-25]使用统计学习和元建模方法预测时钟功率和时钟线长; 文献[26]利用 ANN 预测瞬态时钟功耗.

在时钟树的预测和生成方面, 文献[27]提供了一种完整解决方案, 如图 6 所示. 首先从触发器分布、时钟线网分布以及试布线中提取设计特征; 然后利用提取的特征和数据库中的时钟树训练回归模型并预测 CTS 结果; 最后由回归模型监督, 利用生成对抗网络(generative adversarial networks, GAN)和强化学习(reinforcement learning, RL)来优化 CTS, 并对成功/失败的 CTS 进行分类.

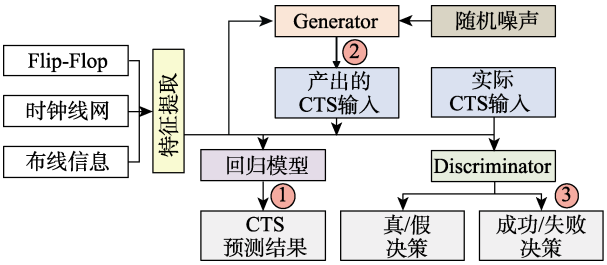


图 6 时钟树生成过程

该工作的主要意义在于提升时钟树预测准确性的同时, 兼顾了 ML 模型的可解释性和通用性等目标.

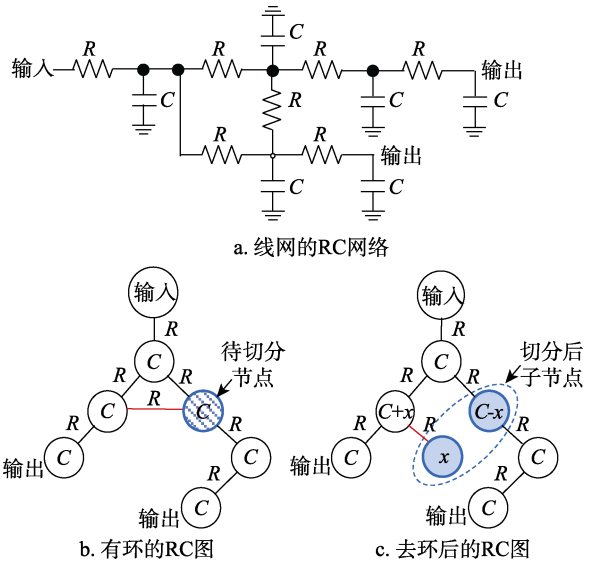
3.4 布线阶段

在布线阶段工具会反复执行时序优化, 调用 Sign-Off 工具进行时序分析和优化将导致运行时间过长, 成为时序优化的瓶颈. 因此, 相对其他阶段, 布线阶段更需要快速、准确的时序预测方法来更好地指导设计中的时序优化, 提高时序收敛效率.

为了缩小互连线时延计算与 Sign-Off 工具的时延偏差, 文献[28]提出基于布线 RC 图信息的互连线时序预测方法.

由于并非所有线网的 RC 图都是无环的树形结构, 对于带环结构, 在没有仿真的情况下环路上延迟传播的方向难以确定. 因此, 文献[28]提出一种非树形结构的构造方法, 即带环 RC 图去环方法. 如图 7 所示, 去环方法步骤如下: 在 RC 图的环路上选择任一节点作为切分节点, 将该节点一分为二, 切分为 2 个不相连子节点, 且 2 个子节点各自连接环路两边, 完成去环; RC 图去环后没有新增边, 所有边对应的电阻值不变, 但去环操作涉及节点的电容值更新, 子节点电容可通过高斯消元法求解; 最后对无环树形结构 RC 图进行时序特征提取.

文献[28]提取的时序特征包含驱动强度、输入 slew 以及 RC 图的相关信息. 其中, RC 图的相关信息包括段数、基于 Elmore 模型的第 1 响应脉冲时延、基于 D2M 模型的第 2 响应脉冲时延、同线网的相邻子树 Elmore 时延、RC 图中从输入节点到目标节点 RC 路径中的电阻以及下游电容和阶段时延



a. 线网的RC网络　b. 有环的RC图　c. 去环后的RC图

图 7　非树形 RC 图采用电容守恒的节点拆分去环方法

等. 最后，采用 XGboost ML 模型训练线时延预测模型. 该工作在布线阶段的 RC 图去环处理以及基于RC树的时序特征提取等方面都进行了充分的研究，且实验应用在工业界电路上的预测准确度高，有较好的实际意义.

3.5　Sign-Off 阶段

在 Sign-Off 阶段可以进行准确的电路参数提取，包括版图布线信息、电容电阻信息等. 因此，基于 Sign-Off 阶段的 STA 预测中，时序特征提取比其他设计阶段更丰富和精确. 同时，Sign-Off 阶段的时序研究问题更多样化.

(1) Sign-Off 工具间一致性问题. 存在多种 STA 商用工具，其时序分析报告往往存在差异. 由于工具许可证费用昂贵，时序分析的运行时间长，因此需要考虑基于已有工具时序报告，实现不同工具间的时序报告预测.

(2) SI 模式时序预测. 由于商业 Sign-Off 工具的非 SI 模式下成本和运行时间都远远低于 SI 模式，而 SI 模式的时序分析更加准确，因此可以考虑利用非 SI 模式的结果预测 SI 模式，在降低成本和缩短时间的基础上提高分析准确度.

(3) 基于路径的时序分析(path-based analysis, PBA). 时序分析策略包括 PBA 和基于图的时序分析(graph-based analysis, GBA). 其中，PBA 更准确但运行时间随着电路规模呈指数级上升；GBA 运行速度快，但存在较大悲观分析. 因此，可以考虑基于 GBA 的结果进行 PBA 预测.

针对上述问题，现有工作采取的改进措施如下.

(1) 缩小不同 Sign-Off 工具分析差异

目前有多个 STA 工具供应商，如 Gary Smith EDA, Synopsys, Cadence, Atrenta, CLK Design Automation, Incentia Design Sytems 和 Mentor Graphics 等，为 IC 设计提供 STA 和 SI 分析工具. 这些商用 STA 工具通常需要较高的许可证费用和较长的运行时间，而且其时序分析报告往往存在差异. 根据需求和产品质量标准，不同 IC 设计公司采用不同的工具作为其 Golden STA 工具. 由于 IC 设计的预算成本限制，可能无法使用特定的 Golden STA 工具，需要将其已有的时序工具结果关联到 Golden STA 结果. 然而，要根据一个工具的报告去预测另一个工具的报告，分析一致性问题十分复杂.

为了最大限度地减少工具差异，实现 Sign-Off 时序工具之间的关联，文献[30]在 SI 模式和非 SI 模式下分别开发了 3 种模型：路径裕量、建立时间和单元延迟；此外，在 SI 模式下还开发了线时延和阶段时延模型，使用分层模型而不是平面模型来改进一致性和减小误差范围；还建立了增量式模型训练方法，以应对新设计出现的异常数据预测问题. 增量式时序模型建立如图 8 所示.

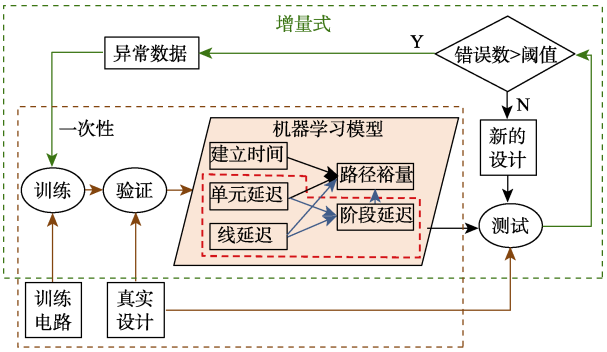


图 8　增量式时序模型建立

(2) 基于非 SI 模式的 SI 模式时序预测

在先进工艺下，信号耦合所引起的增量延迟是一个严重的问题. 通常，IC 设计公司在 SI 模式的 STA 工具许可证上需花费大量资源. 此外，由于算法复杂以及窗口迭代计算等原因，SI 模式的 STA 工具的运行时间很长；而非 SI 模式下的时序分析速度相对要快很多，且许可证成本也比 SI 模式低几倍. 基于非 SI 模式的时序报告，文献[31]开发了基于 ML 的 SI 模式时序预测方法.

商用 Sign-Off 时序工具中，SI 模式和非 SI 模式之间的时序结果往往差异显著. 文献[31]研究了影响 SI 模式中增量延迟的多个电气和电路参数，使用参数特征包括：a. 非 SI 模式下的增量延迟；b. 非 SI 模式下的过渡时间；c. 时钟周期；d. 电阻；e. 耦合电容；f. 耦合与总电容的比率；g. 切换速率

Toggle Rate; h. 攻击者数; i. 受害者线网时序弧出现的阶段与路径中阶段总数的比率; j. 线网 driver 的 Logical Effort; k. 受害者及其最强攻击者在 driver 输出引脚处信号的最大/最小到达时间的差异.

文献[31]使用 ANN 和支持向量机(support vector machine, SVM)[44]进行建模, 并使用混合代理模型(hybrid surrogate modeling, HSM)[24]将来自 ANN 和 SVM 模型的预测值结合起来得到最终预测, 如图 9 所示.

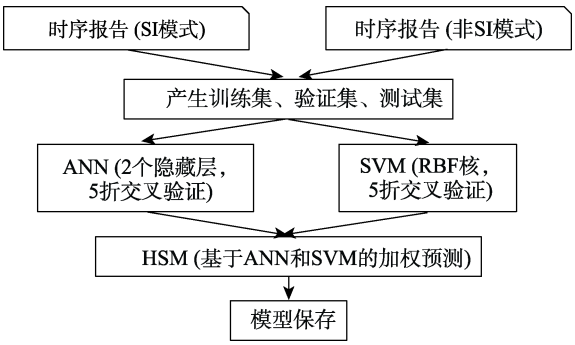


图 9 基于非 SI 模式预测 SI 模式下的时序结果

(3) 基于 GBA 的 PBA 预测

时序分析模式有 2 种: GBA 和 PBA.

a. GBA 模式. 基于时序图找到从起点到终点中每个节点最小的 RAT 和最大的 AT, 节点的 AT 和 RAT 可以分别按节点在时序图中的拓扑顺序和逆拓扑顺序来计算. GBA 可以同时分析所有时序路径, 计算量和运行时间不会随设计规模的增大而大幅增加, 缺点是存在较大的悲观分析.

b. PBA 模式. 基于时序图找到起点到终点的所有路径进行时序分析. PBA 模式可以独立、精确地计算每条路径, 但是随着设计规模的增加, 这种时序分析策略需要单元时延和线时延等多种信息, 增加了计算量和运行时间.

图 10 所示为 GBA 模式和 PBA 模式下 slack 结果对比示例. 可以看出, GBA 模式下, endpoint 的 AT 和 RAT 分别是 11 和 10, slack 为–1, 即判定为关键路径; 而 PBA 模式会分析每条路径的 slack, 只有其中一条路径是关键路径. 相比 GBA 模式, PBA 模式减少了悲观分析程度, 更加接近真实情况, 因此 PBA 模式一般作为 Sign-Off 时序结果. 随着设计复杂度的增加, PBA 的运行时间的增加程度比指数级增长还要快; 而 GBA 模式下结果又太不准确, 文献[32]表明, 最差情况下偏差可达上百皮秒, 分析结果过于悲观. 文献[32]借助 ML 的方法, 基于 GBA 结果预测 PBA 结果.

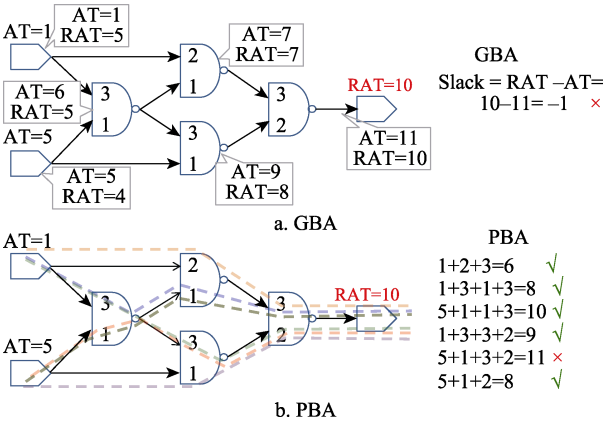


图 10 2 种模式下 Slack 结果对比示例

文献[32]对时序路径采用阶段矫正方式估计 PBA-GBA 偏差. 在时序路径中, n 个连续阶段被称为 n-gram. 随着 n 的增加, PBA-GBA 在时序路径 n-gram 上会产生积累误差. 由于 n>2 的 n-gram 模型会因组合爆炸问题而加重训练的计算负担, 因此采用 2-gram 模型来进行特征提取. 在模型建立阶段, 以 GBA 和 PBA 的时序路径作为输入; 然后针对每个 bigram 单元对提取相关时序特征, 用于模型训练. 在推理测试阶段, 根据新输入的 GBA 时序路径来预测 PBA-GBA 偏差, 从而达到修正时延预测偏差的目的.

3.6 其他相关工作

除了上述 RTL-to-GDSII 流程中的 STA 标量延迟预测方法, 下文简单介绍其他相关工作, 包括统计 STA(statistical STA, SSTA)中的多工艺角延迟预测和时序分析中的 GPU 加速技术.

(1) 多工艺角延迟预测

随着 IC 技术的飞速发展和实际工作环境的多样化, 为了保证芯片运行的稳定性, 必须模拟越来越多的 PVT(process-voltage-temperature)工艺角, 这将消耗大量的设计周期和模拟时间. 因此, 研究人员提出了 SSTA 的概念, 通过概率分布来考虑不同工艺角下的时序变化和波动情况. 针对多 PVT 工艺角的时延预测问题, 文献[33]使用高斯过程回归建模, 通过逐步迭代扩展训练集的方法获得具有最差工艺角的 PVT 子集, 以降低模拟成本. 文献[34]在不涉及跨电压域的延迟预测的情况下, 提出一种使用部分工艺角延迟值来预测其他剩余工艺角延迟值的模型. 文献[3]采用基于反向传播回归的方法预测单个工艺角下的路径延迟波动, 该方法可进一步扩展至未知工艺角的预测. 在宽电压设计中, 文献[36]提出基于双学习技术的时序预测框架, 包括特定电压节点的路径延迟和其他节

点的路径延迟.

(2) 基于 GPU 的时序分析加速技术

由于设计规模越来越大，为加快 STA 的分析效率，可以利用 CPU 的并行性，但一般会受到 CPU 内核的限制，影响加速比. 与 CPU 相比，GPU 在数据密集型的处理上更为强大有利.

文献[37]开发了面向 GPU 的数据结构和内核，以加速 STA 的各种任务，包括层次化、延迟计算和图数据更新. 该算法基于 OpenTimer[45]实现，实验结果表明，其在百万门级电路规模上可高达 3.69 倍的加速. 文献[38]提出基于 GPU 加速的 PBA 算法，在保证精度的同时可以提升几十至上百倍的加速比. 文献[39]针对公共路径悲观移除问题的 HeteroCPPR 算法，实现 CPPR 中的图算法并行加速，实验结果表明，在 4 个 GPU 上可达到 16 倍的加速，在 1s 内处理百万门级电路规模上的 10k~100k 路径生成问题.

与 STA 中计算标量延迟不同，SSTA 主要计算延迟概率分布，得到工艺制造过程变化对延迟的影响. 其中，蒙特卡罗模拟是 SSTA 中重要且耗时的步骤，文献[40]采用 GPU 中单指令多数据流(single instruction multiple data, SIMD)方式执行并行加速，与串行 CPU 实现方式相比，该方法可获得数百倍的加速比. 文献[41]针对主成分分析(principal component analysis, PCA)的 SSTA 开发了基于批处理的任务调度算法，与顺序执行的 SSTA 相比，可提升几十至上百倍的加速比.

4　敏捷设计中的时序预测挑战

敏捷设计的主要目标是实现无人参与的设计自动化，需要依靠各设计阶段性能的准确预测有效地指导各个阶段的设计优化，尽可能减少人工干预以及设计的返工迭代. 随着 IC 技术的发展，工艺特征尺寸不断缩小，芯片集成度与频率不断提高，导致时序收敛是当今芯片设计面临的一个巨大挑战，其影响着产品的设计质量和上市时间. 基于 ML 的时序分析预测可以有效地帮助设计的时序收敛，是实现无人参与敏捷设计的重要研究内容.

时序分析预测的挑战主要体现在以下方面:

(1) 更多的工作场景. 在当前工艺条件下，超大规模设计包含的工作场景越来越多，如何处理这些庞大的时序数据，并在有限时间内做出优化方案，是设计者关注的问题.

(2) 先进工艺的支持. 先进工艺以及低电压芯片设计、工艺的扰动往往呈现非高斯分布的特征，如何构建更加精准的时序模型，也是 EDA 公司和代工厂共同关注的难题.

(3) 保证精度的同时获得更高的分析效率. 主要技术包括多线程、分布式和异构加速的应用，基于 ML 技术的建模与预测，以及层次化 Sign-Off 分析等.

目前，“智能化”的时序预测研究尚处于起步阶段，如何融合体系结构信息、数据采集、ML 建模、与现有工具实用化集成以及预测的可解释性和通用性等问题，仍在探索阶段，其发展方向与趋势也存在着很大的不确定性. 下文就敏捷设计中的时序预测所面临的挑战进行说明.

4.1　融合体系结构信息的时序预测方法

区别于传统设计方法，敏捷设计的上层体系结构设计和后续 RTL-to-GDSII 流程不应是各自独立的，需要为上层体系结构设计和 RTL-to-GDSII 最终结果性能的有效协同提供渠道. 通过利用 RTL-to-GDSII 的经验样本信息进行时序预测建模，使得在体系结构设计阶段，不同的设计方案可以进行快速可行性和性能评估.

敏捷设计 OOA 的松耦合设计方法中，整个系统可以采用高于门级、低于或等于基础模块级的硬件对象为单位进行解耦，每个硬件对象可以有不同的设计方案和参数. 如何对每个硬件对象提取其不同体系结构特征，并采用 ML 方法对不同特征参数的对象组合进行整个系统的时序性能评估，仍需要进一步探索.

4.2　RTL-to-GDSII 流程数据采集

数据集的规模和质量对于预测模型的性能至关重要. 几乎所有 ML 在 EDA 领域的应用都涉及如何产生大规模、合理、干净的数据集.

RTL-to-GDSII 跨越各个设计阶段，如图 11 所示，对于主要设计流程(如逻辑设计、物理布局、时钟树生成、物理布线和 Sign-Off 优化等)而言，每个阶段输出的结果信息不同. 此外，在一般设计流程中，即使同一样本，后续设计阶段还会在前一设计阶段的输出结果的基础上进行调整，如修改逻辑、插入缓冲器等，增加了全流程中样本采集和不同阶段样本匹配的难度. 此外，不同工艺库、不同 PVT 工艺角下，数据样本(如逻辑门和连线等)对应的电容电阻值和时序信息是不同的. 如何在提高样本多样性的同时进行准确的特征选取，对增加时序预测的普适性和实用性十分重要.

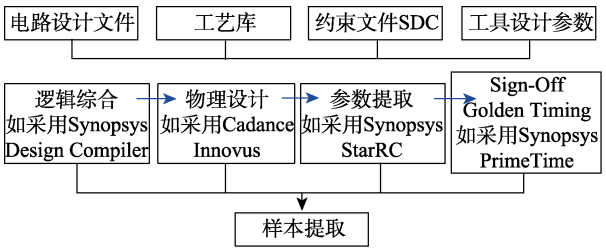


图 11 数据样本产生流程示例

4.3 ML 建模

虽然不同设计阶段都需要调用 STA 进行时序分析，指导该阶段的时序优化和收敛，但是不同阶段提供给 STA 的信息是不同的；在不同的应用场景下，基于 ML 的时序预测目标和问题描述也会有差异. 因此，在不同阶段、不同应用背景下，时序预测建模也需要采取不同的方式.

目前，在时序预测方面，最常用的模型包括线性回归、随机森林、XGBoost 和 GNN 等. 其中，回归模型、随机森林和 XGBoost 等集成学习模型主要针对物理设计及 Sign-Off 阶段应用场景，样本数据一般以单个线网或单个线/单元时序弧等为单位进行预测，而对应的特征信息一般是基于输入的单元、连线和 RC 信息等. 在物理设计之前更早期阶段(如逻辑综合阶段)，由于没有电路的位置信息和走线信息，通常采用 GNN 模型，整个网表可以看成是一张图，每个节点/边对应的特征以及整张图的网表连接关系用于模型训练. 因此，在 GNN 模型中，节点/边不再是单个样本，而是通过图结构互相关联. 在训练过程中，节点/边的视角更加全局化，借此来弥补逻辑综合阶段所缺乏的电路位置和走线等信息.

其他模型，如 ANN，SVM 和卷积神经网络(convolutional neural networks, CNN)等模型也有应用. 文献[31]采用 ANN 和 SVM 进行预测；文献[21]针对逻辑综合阶段的关键路径和非关键路径预判问题，采用 CNN 方法进行二分类.

因此，针对不同的阶段、不同的问题背景和特点，可以利用多种 ML 算法进行建模. 在实践中，需要依据时序预测的具体问题描述找到合适的预测模型.

4.4 与现有 EDA 工具的实用化集成

基于 ML 的时序预测方法需要最终嵌入到 EDA 工具内，指导 EDA 设计优化. 目前，很多 EDA 工具自带时序分析器，如 Innovus 在进行物理设计的过程中，会利用已有的时序分析器进行时序评估和优化. 如何将最新的技术融入到现有流程和工具中，是敏捷设计的重要问题. 根据 ML 时序预测对现有工具设计流程的影响，可以将敏捷设计中无人参与的程度划分成不同的自动化层次.

(1) 层次 1——设置参数约束

该层次的优点是操作简单，可以不改变工具内部实现，利用工具已有的参数设置，通过修改约束文件对时序预测有问题的线网和路径增加额外的约束命令，达到 ML 指导工具进行设计优化的目的. 文献[21]中，在逻辑综合阶段判断路径是否为关键路径，若是，则对该路径从起始节点到终端节点，添加 set_max_delay 约束来指导设计优化.

(2) 层次 2——工具调用时序模型

该层次需要为设计工具提供调用接口和调用命令，保证在设计过程中对设计中间结果进行时序优化指导，这也是目前进行时序优化常见的方式. 由于时序分析工具可能来自不同的开发商，因此大多数情况下将整个网表作为输入进行时序分析，不方便局部的时序分析和修正.

(3) 层次 3——工具与时序模型集成

在设计过程中，设计人员可能需要频繁地对设计改动进行时序预判和评估，集成时序预测模型是最直接的方式，不仅可以全局实时进行时序预测，而且可以增量式对单个线网等改动进行评估. 依靠评估的结果，EDA 工具可以进行灵活的改动以保证后期的时序收敛.

该层次是目前设计工具进行时序分析最直接的模式，但是在 EDA 工具中结合时序模型，最大的困难是需要 EDA 工具丰富的函数接口和数据支持.

(4) 层次 4——时序优化决策

一般时序优化是设计工具基于时序分析结果，做出相关的逻辑修改、线网权重调整、模块重分布和拆线重布等处理方式. 根据不同的时序分析结果给出合理的时序参数输入建议，可以减少设计中人工手动参数设置的不确定因素，降低对设计人员专业性和熟练性的门槛，智能化辅助设计收敛.

4.5 时序预测的可解释性和通用性

与传统的时序分析方法不同，基于 ML 的时序预测方法依赖于经验样本以及外界工艺条件等因素. 对于模型的使用者或设计工具而言，采用 ML 预测是一种“黑匣子”方案，模型的可解释性有待进一步探索. 在通用性方面，如跨工艺库、多工艺角等方面，相比传统时序分析工具仍有许多局限性.

虽然基于 ML 的时序预测方法可以帮助 EDA 设计流程提供有价值的时序评估和优化指导，但是 ML 的广泛应用在很大程度上仍依赖于芯片设

计信任其预测/输出的能力. 例如，基于ML的时序预测方法有可能把实际有时序违例的电路节点、连线或者时序路径预测为没有问题，那么就无法保障后续设计的时序优化和时序收敛. 目前，针对“违例样本”的预测准确度问题，主要通过在预测结果基础上施加悲观拉伸[21]，但不具有广泛实用性. 因此，如何在降低悲观预测的同时保证对违例样本实现 100%的预测，是基于 ML 的时序分析在实用性角度需要解决的关键问题.

此外，在许多时序分析任务中，单纯使用 ML 模型难以满足工业需求，因此 ML 与传统方法的巧妙结合仍具有重要意义. 当前的时序预测辅助 EDA 方法可能仍局限于灵活性较差的设计空间，或旨在解决简化问题. 例如，不同工艺节点下对应的单元和连线特性会有差异，尤其是随着目前特征尺寸已达到几十甚至几纳米阶段，不仅器件尺寸缩小，结构也发生变化. 针对更多的工艺节点和真实场景需求，需要设计新的模型和算法，使 ML 时序模型能够用于实际应用中. 当然，这不仅是时序预测，也是IC设计中所有性能预测都需要面对的挑战.

总而言之，对技术的信任，是基于对其工作原理的理解以及对其安全性和可靠性的评估. 为了信任由算法或 ML 模型做出的决策，设计师或 EDA 工具用户需要评测证明其预测结果是可靠的. 希望看到更多的时序预测以及其他性能预测的研究沿着这条路线发展，让敏捷设计工具值得信赖.

5　结　语

时序预测可以将 Sign-Off 的时序结果前馈到 RTL-to-GDSII 流程各个阶段，指导时序优化，确保时序收敛的同时减少设计迭代次数，缩短设计周期，降低设计成本，因此，准确的时序预测是实现无人干预 RTL-to-GDSII 的时序性能保障. 本文给出敏捷设计下的时序优化的整体框架，并对当前 ML 时序预测方法在设计流程各个阶段(包括逻辑综合、布局、CTS 和布线等阶段)的研究现状进行了综合性阐述. 最后总结了目前ML时序分析所面临的挑战. 作为前沿发展方向，基于 ML 的性能预测评估还有许多未知问题有待探索，需要相关领域学者共同探索，提升现有 EDA 的水平.

参考文献(References):

[1] ITRS. International technology roadmap for semiconductors[OL]. [2022-01-02]. http: //www.itrs2.net/itrs-reports.html

[2] Bao Yungang, Chang Yisong, Han Yinhe, *et al*. Agile design of processor chips: issues and challenges[J]. Journal of Computer Research and Development, 2021, 58(6): 1131-1145(in Chinese)
(包云岗，常轶松，韩银和，等. 处理器芯片敏捷设计方法: 问题与挑战[J]. 计算机研究与发展, 2021, 58(6): 1131-1145)

[3] DARPA. DARPA rolls out electronics resurgence initiative [EB/OL]. [2022-01-02]. https://www.darpa.mil/news-events/2017-09-13

[4] Chen T. An ANN approach for modeling the multisource yield learning process with semiconductor manufacturing as an example[J]. Computers & Industrial Engineering, 2017, 103: 98-104

[5] Ding D, Torres J A, Pan D Z. High performance lithography hotspot detection with successively refined pattern identifications and machine learning[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2011, 30(11): 1621-1634

[6] Ward S, Ding D, Pan D Z. PADE: a high-performance placer with automatic datapath extraction and evaluation through high dimensional data learning[C] //Proceedings of the 49th Annual Design Automation Conference. New York: ACM Press, 2012: 756-761

[7] Wang F C, Zhu H B, Popli P, *et al*. Accelerating coverage directed test generation for functional verification: a neural network-based framework[C] //Proceedings of the 2018 on Great Lakes Symposium on VLSI. New York: ACM Press, 2018: 207-212

[8] Kiely T, Gielen G. Performance modeling of analog integrated circuits using least-squares support vector machines[C] //Proceedings Design, Automation and Test in Europe Conference and Exhibition. Los Alamitos: IEEE Computer Society Press, 2004: 448-453

[9] Kahng A B, Lin B, Nath S. ORION 3.0: a comprehensive NoC router estimation tool[J]. IEEE Embedded Systems Letters, 2015, 7(2): 41-45

[10] Lee Y, Waterman A, Cook H, *et al*. An agile approach to building RISC-V microprocessors[J]. IEEE Micro, 2016, 36(2): 8-20

[11] Chan T B, Kahng A B, Li J, *et al*. Optimization of overdrive signoff[C] //Proceedings of the 18th Asia and South Pacific Design Automation Conference (ASP-DAC). Los Alamitos: IEEE Computer Society Press, 2013: 344-349

[12] Bhasker J, Chadha R. Static timing analysis for nanometer designs: a practical approach[M]. Heidelberg: Springer, 2009

[13] Bodapati S, Najm F N. Prelayout estimation of individual wire lengths[J]. IEEE Transactions on Very Large Scale Integration Systems, 2001, 9(6): 943-958

[14] Fathi B, Behjat L, Rakai L M. A pre-placement net length estimation technique for mixed-size circuits[C] //Proceedings of the 11th International Workshop on System Level Interconnect Prediction. New York: ACM Press, 2009: 45-52

[15] Liu Q, Ma J G, Zhang Q J. Neural network based pre-placement wirelength estimation[C] //Proceedings of the International Conference on Field-Programmable Technology. Los Alamitos: IEEE Computer Society Press, 2012: 16-22

[16] Hu B, Marek-Sadowska M. Wire length prediction based clustering and its application in placement[C] //Proceedings of the 40th Annual Design Automation Conference. Los Alamitos: IEEE Computer Society Press, 2003: 800-805

[17] Kahng A B, Reda S. Intrinsic shortest path length: a new, accurate a priori wirelength estimator[C] //Proceedings of the IEEE/ACM International Conference on Computer-Aided De-

sign. Los Alamitos: IEEE Computer Society Press, 2005: 173-180

[18] Liu Q H, Marek-Sadowska M. Pre-layout wire length and congestion estimation[C] //Proceedings of the 41st Annual Design Automation Conference. Los Alamitos: IEEE Computer Society Press, 2004: 582-587

[19] Ma Y Z, Ren H X, Khailany B, *et al*. High performance graph convolutionaI networks with applications in testability analysis[C] //Proceedings of the 56th Annual ACM/IEEE Design Automation Conference. New York: ACM Press, 2019: Article No.18

[20] Xie Z Y, Liang R J, Xu X Q, *et al*. Net2: a graph attention network method customized for pre-placement net length estimation[C] //Proceedings of the 26th Asia and South Pacific Design Automation Conference. New York: ACM Press, 2020: 671-677

[21] Neto W L, Moreira M T, Amaru L, *et al*. Read your circuit, leveraging word embedding to guide logic optimization[C] //Proceedings of the 26th Asia and South Pacific Design Automation Conference. Los Alamitos: IEEE Computer Society Press, 2021: 530-535

[22] Barboza E C, Shukla N, Chen Y R, *et al*. Machine learning-based pre-routing timing prediction with reduced pessimism[C] //Proceedings of the 56th ACM/IEEE Design Automation Conference. New York: ACM Press, 2019: Article No.106

[23] Kahng A B, Mantik S. A system for automatic recording and prediction of design quality metrics[C] //Proceedings of the 2nd International Symposium on Quality Electronic Design. Los Alamitos: IEEE Computer Society Press, 2001: 81-86

[24] Kahng A B, Lin B, Nath S. Enhanced metamodeling techniques for high-dimensional IC design estimation problems[C] //Proceedings of the Design, Automation & Test in Europe Conference & Exhibition. Los Alamitos: IEEE Computer Society Press, 2013: 1861-1866

[25] Kahng A B, Lin B, Nath S. High-dimensional metamodeling for prediction of clock tree synthesis outcomes[C] //Proceedings of the ACM/IEEE International Workshop on System Level Interconnect Prediction. Los Alamitos: IEEE Computer Society Press, 2013: 1-7

[26] Kwon Y, Jung J, Han I, *et al*. Transient clock power estimation of pre-CTS netlist[C] //Proceedings of the IEEE International Symposium on Circuit and Systems. Los Alamitos: IEEE Computer Society Press, 2018: 1-4

[27] Lu Y C, Lee J, Agnesina A, *et al*. GAN-CTS: a generative adversarial framework for clock tree prediction and optimization[C] //Proceedings of the IEEE/ACM International Conference on Computer-Aided Design. Los Alamitos: IEEE Computer Society Press, 2019: 1-8

[28] Cheng H H, Jiang I H R, Ou O. Fast and accurate wire timing estimation on tree and non-tree net structures[C] //Proceedings of the 57th ACM/EDAC/IEEE Design Automation Conference. Los Alamitos: IEEE Computer Society Press, 2020: Article No.229

[29] Kahng A B, Kang S, Lee H, *et al*. Learning-based approximation of interconnect delay and slew in signoff timing tools[C] //Proceeding of the ACM/IEEE International Workshop on System Level Interconnect Prediction. Los Alamitos: IEEE Computer Society Press, 2013: Article No.106

[30] Han S S, Kahng A B, Nath S, *et al*. A deep learning methodology to proliferate golden signoff timing[C] //Proceedings of the Design, Automation & Test in Europe Conference & Exhibition. New York: ACM Press, 2014: Article No.260

[31] Kahng A B, Luo M L, Nath S. SI for free: machine learning of interconnect coupling delay and transition effects[C] //Proceedings of the ACM/IEEE International Workshop on System Level Interconnect Prediction. Los Alamitos: IEEE Computer Society Press, 2015: 1-8

[32] Kahng A B, Mallappa U, Saul L. Using machine learning to predict path-based slack from graph-based timing analysis[C] //Proceedings of the 36th International Conference on Computer Design. Los Alamitos: IEEE Computer Society Press, 2018: 603-612

[33] Shoniker M, Oleynikov O, Cockburn B F, *et al*. Automatic selection of process corner simulations for faster design verification[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 37(6): 1312-1316

[34] Kahng A B, Mallappa U, Saul L, *et al*. “Unobserved corner” prediction: reducing timing analysis effort for faster design convergence in advanced-node design[C] //Proceedings of the Design, Automation & Test in Europe Conference & Exhibition. Los Alamitos: IEEE Computer Society Press, 2019: 168-173

[35] Guo J J, Cao P, Sun Z, *et al*. Novel prediction framework for path delay variation based on learning method[J]. Electronics, 2020, 9(1): 157

[36] Cao P, Bao W, Guo J. An accurate and efficient timing prediction framework for wide supply voltage design based on learning method[J]. Electronics, 2020, 9(4): 580

[37] Guo Z Z, Huang T W, Lin Y B. GPU-accelerated static timing analysis[C] //Proceedings of the 39th International Conference on Computer-Aided Design. New York: ACM Press, 2020: Article No.147

[38] Guo G, Huang T W, Lin Y B, *et al*. GPU-accelerated path-based timing analysis[C] //Proceedings of the 58th ACM/IEEE Design Automation Conference (DAC). Los Alamitos: IEEE Computer Society Press, 2021: 721-726

[39] Guo Z Z, Huang T W, Lin Y B. HeteroCPPR: accelerating common path pessimism removal with heterogeneous CPU-GPU parallelism[C] //Proceedings of the IEEE/ACM International Conference on Computer-Aided Design. Los Alamitos: IEEE Computer Society Press, 2021: 1-9

[40] Gulati K, Khatri S P. Accelerating statistical static timing analysis using graphics processing units[C] //Proceedings of the Asia and South Pacific Design Automation Conference. Los Alamitos: IEEE Computer Society Press, 2009: 260-265

[41] Shen Y R, Hu J. GPU acceleration for PCA-based statistical static timing analysis[C] //Proceedings of the 33rd IEEE International Conference on Computer Design. Los Alamitos: IEEE Computer Society Press, 2015: 674-679

[42] Karypis G, Aggarwal R, Kumar V, *et al*. Multilevel hypergraph partitioning: applications in VLSI domain[J]. IEEE Transactions on Very Large Scale Integration, 1999, 7(1): 69-79

[43] Datli A, Eksi U, Isik U. A clock tree synthesis flow tailored for low power[OL]. [2022-01-02]. https://www.design-reuse.com/articles/33873/clock-tree-synthesis-flow-tailored-for-low-power.html

[44] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction[M]. Heidelberg: Springer, 2009

[45] Huang T W, Guo G, Lin C, *et al*. OpenTimer v2: a new parallel incremental timing analysis engine[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2021, 40(4): 776-789