# EDA软件NEXT STA的设计与项目规划

光学与电子信息学院

黄兆邦

M202172926

目录

[EDA软件NEXT STA的设计与项目规划 1](#__RefHeading___Toc808_1569586540)

[I 介绍](#__RefHeading___Toc810_1569586540) 1

前言 1

1 通用数字后端EDA典型软件结构 2

2 未来发展趋势与中国需求 2

3 NEXT STA的设计原则与核心目的 3

[II NEXT STA软件设计 4](#__RefHeading___Toc812_1569586540)

1 基于Web技术的跨平台图形软件开发 4

2 基于Python的命令行和脚本编程支持 6

3 基于C/C++的算法开发优化与Web移植（或调用） 6

4更多优化项及可能性 7

III 实现与发展 7

1 项目规划 7

2 风险，成果与未来 7

IV 总结 8

V 展望 8

参考 8

**摘要**

进入二十一世纪以来，芯片制程的不断升级和芯片开发的持续复杂化让EDA工具在集成电路设计中变得不可或缺， EDA市场在可预见的未来还将随着集成电路产业的快速发展而持续增长。目前全球数字后端EDA主要由Synopsys，Cadence，Mentor三家实现垄断，国内EDA工具相较之仍有差距。为缓解中国国内EDA人才和集成电路产业人才短缺的特定问题，本文将突破传统EDA研究集中于功能算法角度的局限，从软件总体设计结构角度出发，提出一种创新性的软件工程结构，强调面向用户及跨平台能力、高度模块化和智能一体化，在静态时序分析功能的基础上实现功能可拓展，为降低EDA工具使用成本和行业壁垒提供软件设计层面的解决方案。

为此，本文主体将分为三个部分进行阐述：其一是当前数字后端EDA工具的软件的结构分析，EDA工具软件技术的未来以及中国独特的EDA工具需求；其二，详细介绍NEXT STA基于Web技术的图形化操作界面、C/C++算法功能实现与Python命令行控制的EDA工具软件工程设计，探讨其技术实现细节及技术栈优劣，验证可行性，并提出具体的设计实现方案；第三，结合一、二部分内容，阐述该EDA软件可能的实现流程规划、最终效果以及未来发展。

关键词：数字后端，EDA，静态时序分析（STA），软件设计，软件架构。

## I 介绍

**前言**

高高在上的行业壁垒让数字后端EDA成了Synopsys、Candence、Mentor Graphic的天下，一批又一批国内优秀学子期待进入这些待遇丰厚的外企工作，国内EDA公司在人员、技术、资金等关键方面明显落后，步步追赶，但尚难以企及。当国内数字后端EDA企业还在基础算法和功能实现方面嗷嗷待哺时，Synopsys已经使用全流程EDA工具和人工智能技术实现“无人参与”芯片的设计到数字后端到流片，差距如此现实。

然而，三大厂商并不是无懈可击，因为他们的数字后端EDA产品绝非完美无缺。在数字后端EDA领域，算法与功能一直是人们研究的重点，软件架构设计却长期不受重视，学习使用EDA工具，面临着安装使用难、平台适应难、资料查找难等一系列让数字后端更趋小众和晦涩的难题，这些难题是当前数字后端EDA工具的壁垒，也是新时代新势力EDA工具发展的机遇，在庞大的垄断巨兽反应过来并花费大量的时间调整他们庞大身躯向正确之未来行进之前，尚有机会绝地反击，争取并驾齐驱之地位。

将数字后端EDA工具软件的设计开发视为一个综合体，强大的功能我们要学习，落后的地方我们要创新，这是辩证的态度。建设一个跨平台面向用户友好的软件平台，让数字后端EDA前所未有的易用、好用、乃至人人都可以学会使用；建设一个软件技术融合平台，让数字后端EDA更加强大，轻而易举融合人工智能、数据分析等技术；建设一个模块化可拓展的开放包容平台，让功能累叠堆砌，逐成万日之工。基于最新的、全栈的软件技术，这些设想并不是遥不可及，本文卑拙，乃尝试窥其雏形，探其先机。

**1 通用数字后端EDA典型软件结构**

当前一般使用的数字后端EDA工具，以Synopsys系列产品为例，通常采取图形化操纵界面 + 命令行脚本支持的形式，限于在linux操作系统上使用，一般仅支持使用CPU进行运算。从软件结构的角度进行分析，可以分为三个主要部分：

* 面向用户的图形化操作界面和可视化操作平台，称之为图形界面模块。
* 支持命令行和脚本运行的软件接口，主要支持TCL/TK，称之为命令行支持模块。
* 基于C/C++实现的软件核心功能和底层算法，称之为算法功能模块。

其中，算法功能模块被视为核心，算法设计和功能设计是一般EDA软件设计开发的重点，而基于TCL的命令行则为数字后端EDA提供了灵活的操作性和强大的脚本复用能力。

从经验的角度来看，当下通用数字后端EDA软件所采取的软件工程工程结构，能较好地满足集成电路领域的发展需求，实现简单到复杂的一系列集成电路设计的物理实现，这主要得益于数字后端EDA工具在算法方面的持续完善和不断优化，强大的软件功能和丰富的智能算法让EDA工具在数字后端领域变得举足轻重，不可或缺。

然而，绝大多数通用的数字后端EDA软件，都背负了沉重的历史包袱，尤其在面向用户角度缺乏考虑，表现在以下几个方面：

* 跨平台安装使用困难。Menter Graphic Calibre在windows环境下安装易出现问题，Synopsys DC、Synopsys ICC2、Candence Innovus等主流的数字后端EDA工具软件都不支持Windows环境，Synopsys系列产品以至于不支持Ubuntu只支持CentOS和RedHEL发行版。
* 支持的脚本编程语言和命令行控制语言落后。TCL/TK运行性能较差，适用范围狭窄，且功能多局限于工具软件领域，难以实现数据分析、数据处理和人工智能的融合应用。
* 完全封闭式的软件管理和自由度极低的功能拓展能力。用户付费使用工具，其工作局限于设置工具本身参数和运行方式，无法对定制化或不理想的功能进行调整，虽可使用TCL/TK拓展编程，但仍无法触碰算法功能核心。

综上作者认为，当前数字后端领域通用的EDA工具，功能足够强大，但在适用性、易用性和可拓展性等方面欠缺面向用户的考虑，造成学习使用EDA工具困难重重。

**2 未来发展趋势与中国需求**

未来数字后端EDA工具的发展值得重视。结合行业总体发展、领头垄断企业的技术判断，以及普遍适用的软件技术趋势，高度总结有以下几个方面：

* 流程阶段的智能化实现。数字后端流程操作具有高度重复性，且所有具体功能都已由EDA工具本身实现，用户所做的设置可不可以也由EDA工具智能替代？[三星](https://baijiahao.baidu.com/s?id=1708035950228169660&wfr=spider&for=pc)最近就做出了首次尝试，使用Synopsys AI软件完成芯片整体设计，后端工程师几个月的工作量和调试调优在强化学习等人工智能技术面前几天就可以完成。
* 基于更强大、更通用的脚本语言控制提升功能性和可拓展性。Synopsys和Cadence将Python视为未来的TCL，目前已有Python库Cocotb能将Cadence，Mentor Graphic等系列的验证EDA工具进行联合仿真。
* EDA一体化设计。Synopsys Fusion Compiler这一一体化EDA平台是数字IC领域EDA工具加快整合的典范。

EDA工具的未来发展还有无限可能，如统一数据库平台等等，而中国集成电路产业对EDA软件的现实需求却是独特而又近在眼前。自中美贸易冲突以来，国内集成电路产业得到了国家和基本市场的重点关注和大力投入，在快速发展的过程中也暴露出了各种问题，其中，各领域专业人才的紧缺反复被提及，人才培养被视为中国集成电路更好更快发展的重中之重。正是由于这种独特的紧迫性，集成电路产业的必由之路 – EDA工具拥有不同于一昧强调功能的巨大需求，中国国内亟需一套面向用户友好、学习成本低廉、实现功能强 大的EDA 软件，从而降低人们接触和使用集成电路EDA工具的成本和难度，然而，当前的数字后端EDA软件，其适用操作系统受众小、使用难度高，采用的TCL编程语言受众小、学习性价比低，加之IC产业本身具有知识壁垒，使得绝大多数人望而却步，即使是集成电路产业的相关人员，初次接触数字后端EDA工具的使用也会感到无从下手，需要花费大量时间从操作平台开始学习，行业内人才流动尚且如此苦难，社会层面的人才开源更仿佛是天方夜谭。

**3 NEXT STA的设计原则与核心目的**

结合介绍部分以上内容，高度总结NEXT STA基于现实原则尝试解决的核心问题：

* 提出一种全新的，适用于集成电路EDA开发和用户使用的软件结构。该软件结构重点在于解决传统EDA工具使用困难、平台单一、可拓展性差、用户受众小的问题，设计更加简洁现代化的图形界面，同时基于Python编程语言进行命令操作和脚本控制，提供桌面端乃至移动端跨平台能力。
* 基于模块化编程首先实现快速有效的STA算法，再基于高度可拓展性开发其它相关功能，全部功能通过API形式层层模块化封装，打下一体化工具的雏形。

## II NEXT STA软件设计

Next STA在软件结构设计方面同样由图形界面模块、命令行模块和算法功能模块实现，但与一般的数字后端EDA工具有如下不同及优势：

* 基于Web的GUI图形界面，面向用户体验更加良好，具备强大的跨平台和开箱即用能力。
* 基于Python的命令行控制和脚本支持，更加易用，更加强大。
* 一整套cutting edge的先进技术栈，将Web图形程序、Python控制台与C/C++实现的算法功能结合，模块化程度高，可拓展能力强。

基于面向用户的设计原则和面向未来的模块化和可扩展性设计，NEXT STA在极大便利用户操作使用EDA工具的同时，还能够提供一套更加包容开放的软件平台，为未来一体化、智能化的全流程设计打下基础，总体结构如图1。

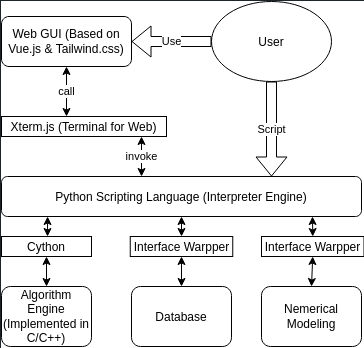


图1 NEXT STA的总体结构

软件设计和架构优势将通过本章后面的三节内容进行详细介绍。

**1 基于Web技术的跨平台图形软件开发**

跨平台软件自操作系统诞生就一直广受关注，如今早已成为现实。QT作为C++阵营最具代表性的软件开发框架，在性能方面出类拔萃，已经被众多CAD、CAE、EDA等工具软件所使用；TCL语言拓展的图形化软件开发平台TK，功能丰富，使用简单，实用性强，Python都针对TK实现了官方支持的库包从而提供更高级的图形化开发功能。以上两门语言，在性能和功能方面各有长处，是开发EDA工具的重要选项。

近年来，随着Web前端技术在应用开发上的大获成功，以Electron为代表，基于网页前端技术的跨平台软件开发成为越来越多开发人员的选择。Web前端技术在跨平台方面具备以下优势：

* 显著的开发周期和维护成本优势。
* 强大的全局功能，Web前端技术涵盖了应用软件开发所需要的绝大部分功能。
* 易于实现软件模块化和插件管理。
* 面向用户的技术哲学，包括美观的应用界面和简洁实用的功能设计。
* 强大的兼容性和众多开箱可用的功能模块。
* 广泛的人才储备和较低的学习成本。

目前，基于Web的桌面端软件的典型代表就是VS Code。自17年发布起，VS Code凭借基于Electron的Web跨平台开发技术在Windows、Linux和MacOS上都可以使用，且简单易用，功能强大，插件丰富，迅速超越Eclipse、JetBrain等IDE平台，短短几年就成为最受欢迎的软件开发工具。VS Code的成功，是面向用户的Web跨平台技术成功挑战传统工具软件市场的成功案例；此外，EasyEDA（图2）已经使用Web技术实现EDA功能，使其一举成为发展速度最快的EDA产品之一，跑在网页上的EDA工具已经不是梦想。

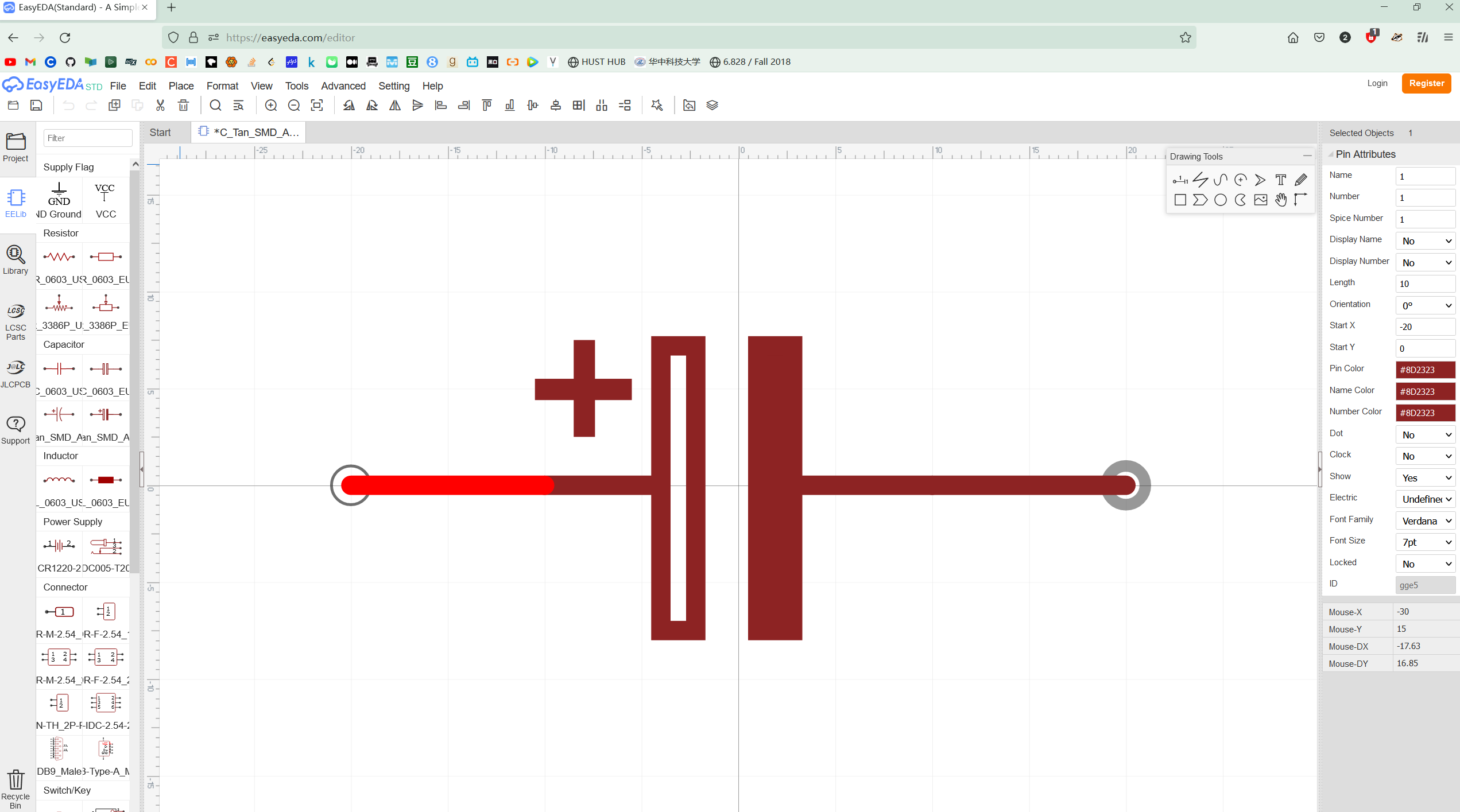


图2 跑在Web端的EDA工具平台 – EasyEDA

然而，Web平台技术也不是有功无过，相对较弱的软件性能、更大的软件体积以及较多的硬件资源耗费，让更加注重性能要求的计算机辅助和自动化工具厂商尚不具备花费巨大资源实现代码迁移的决心。但同样基于Web的Tauri和Neutralino.js的出现，极大改善了Electron慢运行、大体积和高耗费的不足，从图3中我们可以发现，Tauri和Neutralino.js指数级降低了应用体积和内存消耗，使用Web跨平台技术开发数字后端EDA软件工具的基本条件已经完备，可以说只待人临门一脚，开创先河。

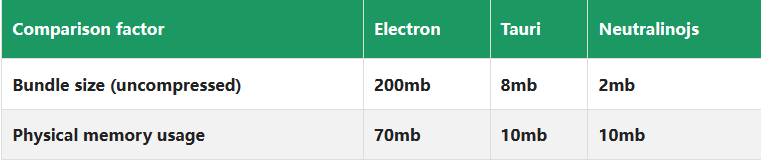


图23基于Web的应用开发框架Tauri与Neutralino.js的显著性能提升

综上，NEXT STA具体采用的技术栈如下： Tauri/Neutralino – 跨平台软件开发框架，

Vue.js – 前端开发框架，

Tailwind.css – 前端渲染框架，

Vite – 前端编译打包框架。

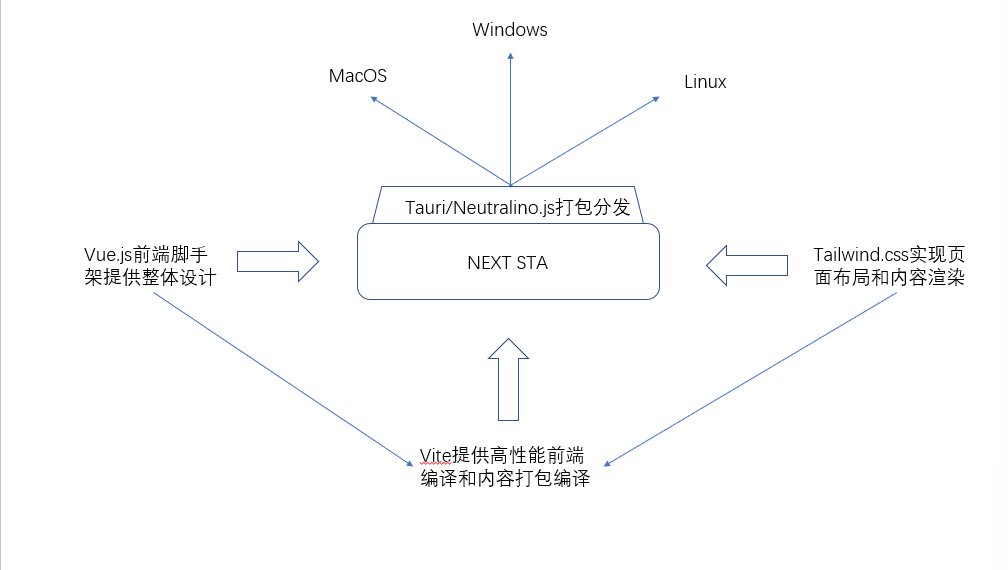


图4 NEXT STA基于Web的客户端/网页端开发

**2 基于Python的命令行和脚本编程支持**

NEXT STA的命令行和脚本支持，不再使用传统的STA，而是基于Python。Python能够实现以下TCL望尘莫及的技术优势：

* 大数据分析和数据可视化分析。通过Python可以调取NEXT STA中的各种参数，包括Cell的延时、面积、RC值等，再使用Pandas、Numpy、Plotly等Python库自由进行数据分析并将他们可视化，从而直观反映当前软件的运行状态和时序分析结果，极大方便耗费统计和频率分析。
* 使用人工智能技术处理数据并实现自动化。结合Python语言在人工智能领域的广泛运用，轻松将Pytorch、Tensorflow等框架运用到任何你想进行优化和自动化处理的地方，或者使用机器分类和强化学习等手段进行算法优化。
* 使用Python和NEXT STA提供的数据做任何Python能做到的事情，包括将NEXT STA部署到本地或服务器上远程访问。

NEXT STA 将使用Xterm.js将Python终端与GUI图形界面进行整合，同时采用Boost.Python/Cpython/Pybind11技术实现C++程序或包的Python封装和调用。

**3 基于C/C++的算法开发优化与Web移植（或调用）**

NEXT STA整体表现除了基于Web的GUI之外，更重要的是功能引擎的性能，其将基于C/C++来编写，以追求高性能。

功能引擎将实现图分割、路径搜索、时序分析等算法，每个算法功能的实现都是一个模块，整个STA算法是一个大的模块，高度模块化将带来以下显著优势：

* 用户可自定义，从而实现插件化管理。结合上节使用Python命令行，用户可以用Python实现自己想实现的算法或者功能，替换掉原有的功能模块或者锦上添花，或者使用C/C++语言实现，然后再使用Python进行调用。所有的这些模块，无论是内置的功能还是用户实现的功能，都可以视作插件，从而进行复用和管理。
* 强拓展性和整合后易实现一体化。每个模块只负责端到端的处理，模块层层叠加，就可以不断提高端到端处理的距离，从而实现一体化软件设计和流程阶段的自动化处理。

所有的C/C++功能模块都将通过Python命令行来进行调用，Python将C/C++实现的功能和使用的数据通过面向对象的class进行封装。

**4更多优化项及可能性**

* NEXT STA的数据可视化功能，基于SVG.js和WebGL实现。
* Python、C/C++以及Javscript之间的数据转换将基于Cython实现。

**III 实现与发展**

**1 项目规划**

* NEXT STA的核心是使用Python命令行控制C/C++实现的功能核心，所以算法功能的实现和Python命令行系统的构建是重中之重，也是开发的优先选项，模块化、可拓展性是关注重点，其次Python和C/C++语言的接口模块也很重要。
* 核心部分搭建实现后，将使用Web技术设计面向用户的GUI界面，完成前端设计，可以先提供网页端服务，确保GUI界面稳定性后再将Python命令行和功能引擎封装，进行跨平台打包分发。
* 数据可视化等功能将作为单独模块嵌入GUI图形开发过程中，只与功能引擎进行数据交流。

NEXT STA还将不只是一个静态时序分析工具，NEXT将会是一个崭新的软件平台，是包容而稳固的地基，而STA只是项目成型所选择的单一功能。当软件开发基本并逐渐稳定后，更多的、更先进的算法功能都将作为一个个砖块加入并累积起来，最终形成一座高楼大厦。

未来更好体现开放包容的思想，同时加速功能开发和软件规模化的速度，成型后的NEXT STA可能开源，吸引更多的工程师和用户共建共享。

**2 风险，成果与未来**

从零设计、全栈开发一套功能稳定，同时包括图形界面和命令行控制系统的数字后端EDA工具绝非易事，面临着巨大挑战，但也蕴含着丰硕成果的可能。数据跨编程语言的交流、文档系统和异常处理机制、命令行和图形界面的设计，这些都是更多更大的挑战中的一部分。但是需要看到，若能克服这些挑战那必将是硕果累累，会有一个可以通过网页进行访问、在各个操作系统上都能轻易使用的EDA工具，这个EDA工具将功能模块化和流程一体化嵌入到了骨子里，所有的功能都将基于后端流程，通过一个个C/C++的模块进行实现，再由Python这一极为简单但又强大的脚本语言进行组装和调用；用户可以使用Python进行数据统计、数据分析、引入人工智能算法进行定制化操作，而当除了功能模块足够多时，NEXT STA将不只是STA静态时序分析工具，而是一个可以在流程级别完成自动化实现的智能工具和开放平台，从而为EDA助益集成电路产业发展提供无限可能。

**IV 总结**

本文综合当前数字后端后端EDA工具的不足和优势，结合EDA一体化、智能化的未来趋势，重点考虑国内集成电路产业对EDA工具易学易用的强烈需求，抛弃了传统EDA工具基于TCL/TK让用户功能参数设置和功能粘合的一般模式，引入最先进的软件技术和开发理念，提出了一种在平台兼容、用户体验、功能拓展等关键方面都具有极大提升的软件架构，具备原创性和创新性。

**V 展望**

NEXT STA绝不会是一个设想，亦或只是行走在纸上的架构，作为华中科技大学光学与电子信息学院软件工程专业的学生，我深刻感受到了软件工程在集成电路EDA产业中的巨大作用和应用潜力，在今后的研究生生涯中，我还将以设计开发实现新时代新特色的数字后端EDA工具为指引，用一行一行扎实地代码将NEXT STA带入现实世界之中。

共勉！

**参考**

[1] [LinkedIn - Python is all set to disrupt HW verification](https://www.linkedin.com/pulse/python-all-set-disrupt-hw-verification-avidan-efody).

[2] Cocotb: a Python-based digital logic verification framework, Ben Rosser, University of Pennsylvania, December，11, 2018.

[3] [EDA-Solution – Choosing a programming/automation language](https://www.eda-solutions.com/tn020/).

[4] [未来智库 – 集成电路EDA产业研究：国产EDA披荆斩棘，乘风崛起](https://baijiahao.baidu.com/s?id=1710143151736487595&wfr=spider&for=pc).

[5] 朱琪凯, 章珺. 集成电路设计EDA发展现状及其未来[J]. 科研, 2017(2):00049-00049.

[6] [中国电子报 – 未来五年将是我国集成电路产业发展的机遇期](http://www.elecfans.com/bandaoti/eda/202007271258839.html).

[7] [Zrix - The Future of Cross-Platform App Development](https://www.zrix.com/blog/future-of-cross-platform-app-development).

[8] [Cython Source Site](https://github.com/cython/cython).

[9] [Pybind11 Source Site](https://github.com/pybind/pybind11).

等