

PaperPass旗舰版检测报告

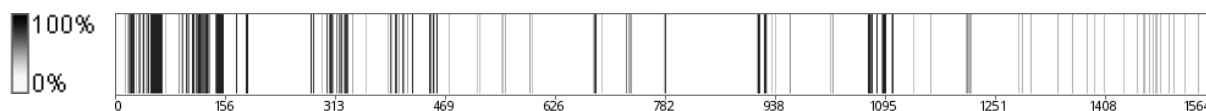
简明打印版

比对结果(相似度):

总体: 22% (总体相似度是指本地库、互联网的综合对比结果)
本地库: 17% (本地库相似度是指论文与学术期刊、学位论文、会议论文、图书数据库的对比结果)
期刊库: 6% (期刊库相似度是指论文与学术期刊库的对比结果)
学位库: 14% (学位库相似度是指论文与学位论文库的对比结果)
会议库: 2% (会议库相似度是指论文与会议论文库的对比结果)
图书库: 6% (图书库相似度是指论文与图书库的对比结果)
互联网: 11% (互联网相似度是指论文与互联网资源的对比结果)

编号: 5AC9CC0F648CC5LK6
版本: 旗舰版
标题: 基于机器学习的28nmSRAM编译器表征预测
作者: 余一奇
长度: 38228字符(不计空格)
句子数: 1564句
时间: 2018-4-8 16:00:15
比对库: 学术期刊、学位论文、会议论文、书籍数据、互联网资源
查真伪: <http://www.paperpass.com/check>

句子相似度分布图:



本地库相似资源列表(学术期刊、学位论文、会议论文、书籍数据):

- 相似度: 3% 篇名: 《超低能耗亚阈值SRAM电路设计》
来源: 学位论文 东南大学 2011
- 相似度: 2% 篇名: 《人脸识别算法研究和实现》
来源: 学位论文 北京交通大学 2007
- 相似度: 2% 篇名: 《加权多宽度高斯核及其支持向量分类和网络核模式》
来源: 学位论文 中国海洋大学 2009
- 相似度: 2% 篇名: 《基于支持向量机的线虫蛋白质相互作用判别》
来源: 学位论文 中山大学 2009
- 相似度: 2% 篇名: 《基于机器学习方法的人脸识别研究》
来源: 学位论文 上海交通大学 2008
- 相似度: 2% 篇名: 《人工智能及其应用》
来源: 书籍数据 清华大学出版社 2003-9-1
- 相似度: 2% 篇名: 《基于支持向量机的蛋白质一级序列分类方法研究》
来源: 学位论文 华中科技大学 2005
- 相似度: 2% 篇名: 《概念学习中Find-S算法和后选删除算法的比较研究》
来源: 学位论文 河北大学 2004
- 相似度: 2% 篇名: 《人工智能及其应用研究生用书》
来源: 书籍数据 清华大学出版社 2004-8-1
- 相似度: 2% 篇名: 《人工智能及其应用》

- 来源：书籍数据 清华大学出版社 1996-5-1
11. 相似度：2% 篇名：《基于CloudStack云平台的神经网络分类器的设计与实现》
来源：学位论文 北京交通大学 2015
12. 相似度：2% 篇名：《基于28nm工艺低电压SRAM单元电路设计》
来源：学位论文 安徽大学 2017
13. 相似度：2% 篇名：《现代信息技术》
来源：书籍数据 电子科技大学出版社 2007-11-1
14. 相似度：2% 篇名：《黑龙江省医疗气象预报系统的研究与实现》
来源：学位论文 哈尔滨工程大学 2011
15. 相似度：1% 篇名：《人工智能原理及应用》
来源：书籍数据 南海出版公司 2006-2-1
16. 相似度：1% 篇名：《面向Internet信息服务的研究》
来源：学位论文 浙江大学 2001
17. 相似度：1% 篇名：《无线接收机中高动态范围数字自动增益控制电路设计》
来源：学位论文 西安电子科技大学 2011
18. 相似度：1% 篇名：《基于小波变换的膜蛋白跨膜螺旋区段预测研究》
来源：学位论文 上海大学 2005
19. 相似度：1% 篇名：《机器人技术与应用》
来源：书籍数据 清华大学出版社 2006-9-1
20. 相似度：1% 篇名：《深亚微米标准单元库设计与优化》
来源：学位论文 西安电子科技大学 2010
21. 相似度：1% 篇名：《40nm可制造标准单元库的设计与实现》
来源：学位论文 湖南大学 2012
22. 相似度：1% 篇名：《基于Memory Compiler实现0.25微米高性能SRAM之设计方法》
来源：学位论文 上海交通大学 2007
23. 相似度：1% 篇名：《基于强化学习的神经网络在船模速度控制中的应用》
来源：学术期刊《计算机时代》2009年4期
24. 相似度：1% 篇名：《强化学习理论、算法及应用》
来源：学术期刊《控制理论与应用》2000年5期
25. 相似度：1% 篇名：《基于多层强化学习的多机器人团队协作》
来源：学位论文 上海交通大学 2006
26. 相似度：1% 篇名：《多智能体系统的学习策略研究》
来源：学位论文 广东工业大学 2009
27. 相似度：1% 篇名：《基于支持向量机优化RBF神经网络的算法及应用研究》
来源：学位论文 江西理工大学 2008
28. 相似度：1% 篇名：《移动机器人局部路径规划方法的研究》
来源：学位论文 济南大学 2008
29. 相似度：1% 篇名：《一种用于锁相环的压控振荡器的设计》
来源：学位论文 华中科技大学 2005
30. 相似度：1% 篇名：《基于强化学习的机器人操作手轨迹规划方法研究》
来源：学术期刊《制造技术与机床》2005年5期
31. 相似度：1% 篇名：《湿法烟气脱硫系统节能减排的优化设计》
来源：学位论文 华北电力大学 2013
32. 相似度：1% 篇名：《受限玻尔兹曼机分类算法的改进》
来源：学位论文 厦门大学 2017
33. 相似度：1% 篇名：《微博用户关注网络结构研究及用户影响力的实证分析》
来源：学位论文 首都经济贸易大学 2017

互联网相似资源列表：

1. 相似度：3% 标题：《工艺角 (Process Corner) PVT(P...》
http://blog.sina.com.cn/s/blog_5b07397f01011c4d.html
2. 相似度：3% 标题：《[转载]工艺角, PVT, TT, SS, FF, FS...》
http://blog.sina.com.cn/s/blog_60fdc06c0102vygf.html
3. 相似度：3% 标题：《corner资料》
<http://www.docin.com/p-237060693.html>

- 4.相似度: 3% 标题:《半导体工艺角_百度文库》
<https://wenku.baidu.com/view/ad81700c2e3f5727a5e962f2.html>
- 5.相似度: 3% 标题:《工艺角 corner_百度文...》
<https://wenku.baidu.com/view/bcbe0efa941ea76e58fa044c.html>
- 6.相似度: 3% 标题:《超低能耗亚阈值SRAM电路设计 - jz.doc...》
<http://www.docin.com/p-1144395921.html>
- 7.相似度: 3% 标题:《工艺角 Process Corner 浮£云...》
<http://www.docin.com/p-382457301.html>
- 8.相似度: 3% 标题:《工艺角 Process Corner 浮£云...》
<http://www.doc88.com/p-241584026289.html>
- 9.相似度: 2% 标题:《Python粉都应该知道的开源机器学习框架: Sc...》
<https://www.leiphone.com/news/201701/ZJMTak4Y8ch3Nwd0.html>
- 10.相似度: 2% 标题:《一个开源机器学习框架新手入门, Scikit-le...》
http://www.eefocus.com/wjwzl/blog/17-01/403671_7ac7e.html
- 11.相似度: 2% 标题:《Scikit-Learn实战KNN》
<http://www.mamicode.com/info-detail-1786672.html>
- 12.相似度: 2% 标题:《Scikit-Learn实战KNN-布布扣-b...》
<http://www.bubuko.com/infodetail-2051034.html>
- 13.相似度: 2% 标题:《Python粉都应该知道的开源机器学习框架: Sc...》
<https://www.leiphone.com/news/201701/ZJMTak4Y8ch3Nwd0.html?from=timeline&viewType=wxixin>
- 14.相似度: 2% 标题:《半导体工艺角-土地公文库》
<http://wenku.todgo.com/jingguanyingxiao/17067333bbc12.html>
- 15.相似度: 2% 标题:《科普 如何通过资料检索戳穿PCEVA枪文 - ...》
<http://www.pc426.com/thread-63280-1-1.html>
- 16.相似度: 2% 标题:《大数据测试学习笔记之Python工具集 - CS...》
<https://blog.csdn.net/zhusongziye/article/details/79560189>
- 17.相似度: 2% 标题:《——关于超低能耗亚阈值SRAM电路设计》
<http://www.doc88.com/p-9572606381269.html>
- 18.相似度: 2% 标题:《机器学习中有监督学习——回归 - 机器学习知识库...》
<http://lib.csdn.net/article/machinelearning/1130>
- 19.相似度: 2% 标题:《机器学习有监督学习之--回归 - CSDN博客》
<https://blog.csdn.net/zkl99999/article/details/45726479>
- 20.相似度: 2% 标题:《监督学习(机器学习)》
http://www.360doc.com/content/16/0222/14/24811_536429749.shtml
- 21.相似度: 1% 标题:《机器学习有监督学习之--回归 - 博临天下 - ...》
<https://www.cnblogs.com/fanyabo/p/4060498.html>
- 22.相似度: 1% 标题:《线性回归 - 大数据 - 木鱼驿站[ITfish...》
<http://itfish.net/article/56453.html>
- 23.相似度: 1% 标题:《静态时序分析基础(转)_百度文库》
<http://wapwenku.baidu.com/view/64a198ee49649b6648d74767?pu=>
- 24.相似度: 1% 标题:《静态时序分析基础(转)_johanna_新浪博客...》
http://blog.sina.com.cn/s/blog_66fe47530100hv8d.html
- 25.相似度: 1% 标题:《静态时序分析基础(转)》
<http://www.docin.com/p-1761044661.html>

全文简明报告:

摘要

{64%:随着移动互联网的迅速发展, SOC作为便携式智能终端的信息处理能力和待机时间的要求也日趋提高,} 存储器作为 SOC的重要组成部分,一般而言面积占整个 SOC芯片面积的一半以上,尤其在高性能处理器中, SRAM占芯片的面积已经超过80%. 存储器的

性能对SOC芯片的影响很大。

如今的 SRAM设计中，越来越多的低功耗技术和高速技术被提出，比如 DVFS技术、电荷泵技术等，在诸多优化技术带来诸多便利的同时，SRAM的设计周期相应也延长了，尤其在仿真环节中，相比于原本就耗时的SRAM设计仿真周期而言，这些优化技术无疑也增加了仿真周期和设计周期。

本文主要内容和研究对象是：{41%：第一，介绍机器学习，详细阐述了最小二乘法回归算法；} 第二，介绍了标准单元库，分析了十个标准单元的时序，通过电路参数建模实现时序预测和通过时序建模实现时序预测； 第三，介绍SRAM的电路结构、基本原理和特征化，主要对表征中的时序和功耗进行分析，并通过时序建模实现时序预测、按温度划分功耗实现功耗预测； 第四，分析并指出本方法在SRAM设计周期中对于设计周期和仿真周期的提升效果。

关键词：{47%：标准单元库静态随机存取存储器最小二乘法编译器表征预测}

第一章绪论

1.1.1半导体集成电路介绍

{85%：随着移动互联网和物联网的发展，集成电路芯片广泛应用于各种领域，如手机，PC，可穿戴设备等，其中手机的出货量已经超过71亿。} {42%：如今空前辉煌的电子信息产业链，是以六十多年来集成电路的发展为基础的。}

{64%：2017年全球集成电路产品的市场销售额大约在4000亿美元左右，其所撬动的电子信息产业的产值为15万亿美元，已然超过全球GDP的五分之一。}

国际半导体产业协会（SEMI）发布的各家产业研究机构对2018年半导体产业营收的预测增长率为7~8%。 {73%：在全球集成电路发展的浪潮中，我国集成电路行业也有不错的表现，} {93%：中国在智能手机为代表的移动终端的AP、通信模块方面已占据全球大约17%的市场份额，} {90%：但是在应用集成电路的其他领域，国产芯片所占市场比例几乎为零。} {86%：从国内的集成电路规模来看，我国集成电路产业正进入快速发展期，并将以一个近20%的增长率持续增长，2017年产值已达到5000多亿。} {84%：但是国内集成电路仅30%左右是自给，其余依赖于进口，其进口额已然超过石油成为第一大进口商品，这种现状亟待解决。}

{91%：在经济方面，不论是国内还是全球，发展集成电路都有着重要的意义。}

1.1.2存储器及SRAM介绍

图1-1SOC中存储器的使用情况

{42%：存储器作为现在VSLI中重要的组成部分，随着近年来半导体技术的发展，存储器在SoC上占据越来越重要的比重，} {45%：如今存储器面积已经达到SoC的80%，而SRAM作为存储器的重要角色，亦具有十分重要的意义。}

{60%：存储器的种类繁多，分类性质也很多，因为存取方式的不同，可以分为随机存取存储器（Random Access Memory）、顺序存取存储器（Serial Access Memory）和按内容寻址存储器（Content Access Memory）。

{59%：随机存取存储器根据断电后能够继续保存信息而分类，可以分为易失性存储器（VM）和非易失性存储器（NVM）两类。} {87%：通常易失性存储器又分为静态随机存取存

存储器（SRAM）和动态随机存取存储器（DRAM）。}

{70%：SRAM是最广泛应用的存储器之一，主要用两个交叉耦合的反相器存储数据，只要电源不断电，数据就一直被保存。}

{84%：当今SOC处理器的主频已经达到4GHZ，片外大容量主存储器的频率只有600MHZ左右。}

SRAM凭借面积小，容量合适，速度快和很好的鲁棒性等优势，作为SOC芯片的存储设备。

{45%：SRAM的速度可以向逻辑电路那么快，完全可以满足处理器的主频需求，在待机的状态下功耗较低，} {68%：具有更高的带宽，可以满足系统对带宽的需求。} {100%：更好的工艺缩放性，可以随着工艺节点的缩小得到更好的集成度和性能。} SRAM在拥有这些其他存储器不可比拟的优势的同时，也意味着SRAM具备的设计难度和要求。

1.1.3SRAM设计优化技术

{91%：早在1973年，戈登摩尔曾指出，一个硅片上晶体管的数量每隔18个月就翻一番（摩尔定律）。} {49%：在过去的几十年中，集成电路行业的发展一直印证着摩尔定律，一颗芯片上的晶体管数量一直保持大约每两年翻一番。} {54%：随着晶体管数量的剧增，能耗问题、散热问题变得日益严重。} {77%：其中，低功耗已经成为集成电路设计的重要指标。}

{98%：目前工业界常用的低功耗设计手段有门控时钟（Clock Gating）、电源关断（Power Shut-off）、多阈值电压（Multi-Vth）、多电源电压（Multi-Supply Voltage）、动态电压频率调节（DVFS, Dynamic Voltage and Frequency Scaling）、衬底电压偏置（Substrate Biasing）等。{100%：其中门控时钟是较为常用的降低动态功耗的设计手段。} {98%：该设计手段通过使能信号控制时钟传播来降低芯片的动态功耗，它需要引入新的设计单元，} {100%：因此一个信号控制的寄存器数量越多，控制层次也相应越高。} {100%：电源关断则是通过使能信号控制一个区域的电源来降低芯片的泄漏功耗。} {96%：当使能信号无效时，整个区域不供电，因此不产生泄漏功耗。} {100%：需要注意的是，这种设计方法需要引入较多的开关单元和控制单元，} {100%：而且在供电网络上插入控制单元会在使能信号的开关瞬间对周围区域的逻辑器件造成电压波动，} {100%：因此在引入该方法的分析、验证上必须仔细考虑。} {100%：多阈值电压通过将不同阈值电压器件混合使用来降低芯片的泄漏功耗。} {100%：多电源电压则是在芯片的不同区域根据需求选用不同的电源电压值，它可以在降低动态功耗的同时降低静态功耗。} {78%：多电源电压、动态电压调节（DVS）、以及动态电压频率调节技术都是通过调节电源电压达到降低功耗的目的。} {100%：多电源电压是静态电压调整方法，不同的电源电压对应不同的工作性能和功耗。} {100%：DVS和DVFS则是自适应性的方法，他们通过识别处理器的利用率工作：} {100%：当处理器负载较轻时，降低频率和电压；} {100%：当处理器满负荷工作时，系统在最大频率和电压条件下运行。} {96%：这些技术已被广泛应用于商用微处理器，有着显著的节能特性。}

{83%：动态电压频率调整技术（DVFS）是根据芯片所运行的应用程序对计算能力的不同需要，} {74%：动态的调整芯片不同模块的运行频率和电压，从而达到节能的目的。}

文献11中3 D图形处理中有三个独立的模块，每个模块有不同的性能需求，将不同的三个模块应用 DVFS技术可以实现每个功率区域的高效的能耗减少。最终，GPU系统产生相同输出结果的同时，应用DVFS技术的电源管理比正常工作的电源管理实现节省功率约65%。

{40%：在各种层次的低功耗技术广泛应用在 SRAM设计的时候，在带来显著减少功耗的同时，} 也给 SRAM设计提出了相应的附属要求，以 DVFS为例，不同模块动态调整工作电压和工作频率，对于设计的要求，尤其是仿真的要求提高了很多。通常SRAM编译器都

有几千个实例，对于某个实例都要在不同的PVT (Process、Voltage、Temperature) 进行仿真。DVFS使得SOC的电压节点增加了很多。以TSMC 28nm HPCPLUS为例，标准电压是0.9v，相对应的SRAM的签出需要6个PVT的仿真。电压节点从0.72v到1.08v时，以90mv为间隔，一共需要30个corner的签出。

以50个 Intel (R) Xeon的4核 CPU，其频率为3.60 GHZ，以及足够的 EDA licenses，Memory Compiler的表征仿真需要约340小时，基于机器资源和时间成本的考虑，以及更多优化技术的加入，未来 SRAM compiler表征所需的仿真将会呈现阶梯式的增长。本课题的出发点就是提出利用机器学习算法，通过已知的仿真数据构建模型，去预测未知的仿真数据，从而实现节省机器资源和减少时间成本的目的，进而优化整个 SRAM compiler的设计效率。

1.3论文的主要内容

本文主要基于当今微电子行业对于机器学习的应用现状与前景，总结了机器学习在存储器设计中应用的可行性与可靠性，确立了本文的课题：基于机器学习的28nm工艺SRAM Compiler特征化。围绕这一主题，基于 SRAM编译器的设计现状，针对仿真过程的耗时长、占用资源比重大等不足，提出了应用机器学习算法简化仿真，并先后在标准单元库和 SRAM compiler的设计中应用，{40%：以期对未来的机器学习应用于集成电路设计的重要环节中有一定的指导意义。}

本文的主要内容如下：

{44%：第一章为绪论，主要介绍了 SRAM compiler的研究背景、意义，系统阐述了机器学习应用于 SRAM compiler的应用现状和发展前景，} {58%：并对本文的研究方法做了总结概括。}

第二章首先简单的介绍了机器学习及其发展现状与应用前景，主要阐述了回归算法，着重分析了最小二乘法和支持向量机的算法原理和应用场景。

第三章介绍了标准单元库及其时序，并通过支持向量机算法，拟合已有的时序数据，预测未知的时序数据。

{41%：第四章介绍了SRAM compiler的设计流程，着重描述了表征的三个模块：} 时序、功耗和电容，并对三个模块分别使用最小二乘法进行数据的预测。

{70%：第五章为总结与展望，对本文的工作进行了总结，并对本课题的研究进行了展望。}

第二章最小二乘法

2.1机器学习

2.1.1 机器学习的定义

机器学习 (Machine Learning) 是一门人工智能的学科，{83%：它所讨论的是在从有限观察概括特定问题世界模型的机器学习也从有限观察发现观测数据中暗含的各种关系的数据分析的方法，} {87%：其领域的主要研究对象是人工智能，研究热点是如何在经验学习中改善具体算法的性能。} {49%：机器学习主要应用于数据挖掘、语音和手写识别、计算机视觉、生物特征识别、} {81%：自然语言处理、医学诊断、生物特征识别、搜索引擎、医学诊断、检测信用卡欺诈、} {68%：DNA序列测序、证券市场分析、战略游戏和机器运用

等领域。}

2.1.2 机器学习的发展史

{93%：20世纪90年代初期，当时的美国副总统提出了一个重要的计划：} 国家信息基础设施计划（National Information Infrastructure）。 {90%：这个计划的技术含义主要包含了四个方面内容：}

{100%：1．不分时间与地域，可以方便地获得信息。}

{95%：2．不分时间与地域，可以有效地利用信息。}

{100%：3．不分时间与地域，可以有效地利用软硬件资源。}

4．保证信息安全。

信息有效利用问题的本质是： {75%：如何根据用户特定的需求从海量的数据中建立适当的模型或者发现有用的知识。} {94%：对计算机科学来言，这就是机器学习。} {68%：机器学习作为人工智能领域较为年轻的分支，它的发展历程大致上可分为四个时期[40][43]。}

{97%：第一阶段是从20世纪50年代中叶到60年代中叶，属于热烈时期。}

{97%：第二阶段是从20世纪60年代中叶至70年代中叶，被称为机器学习的冷静时期。}

{100%：第三阶段是从20世纪70年代中叶至80年代中叶，称为复兴时期。}

机器学习的最新阶段始于1986年。 {79%：机器学习开启进入新阶段的主要表现在下列几个方面：}

{78%：1．机器学习已成为新兴的热门学科，并在高校形成一门课程。} {72%：它综合了应用心理学、数学、自动化、生物学、神经生理学和计算机科学形成机器学习理论基础。}

{77%：2．结合多种学习方法，以取长补短等多种形式的形成的集成学习系统研究正在兴起。} {94%：特别是连接学习和符号学习的耦合能够更好地解决连续性信号处理中知识与技能的获取与求精问题而受到重视。}

{68%：3．人工智能和机器学习的各种基础问题的统一性的观点正在形成。} {68%：比如问题求解和学习相结合进行、知识表达使得学习的观点发生了通用智能系统的模块学习。} {87%：类比学习和问题求解结合的基于示例方法已成为经验学习的一个重要方向。}

{71%：4．随着各种学习方法的应用范围不断拓宽，一部分方法已经形成了商品。} {97%：归纳学习的知识获取工具已在诊断分类型专家系统中广泛的使用。} {66%：其中连接学习在声图文的识别应用中占优势；} {79%：而分析学习已经用于设计综合型的专家系统；} {56%：另外，强化学习和遗传算法在工程控制应用中有很好的应用前景；} {55%：最后与符号系统耦合的神经网络方法已经在智能机器人运动规划和企业的智能管理中发挥着重要作用。}

5．机器学习的学术会议空前繁多。 {75%：国际上除每年一次的机器学习研讨会外，还有遗传算法会议以及机器学习理论会议等。}

2.1.3 机器学习分类

机器学习有很多算法，分类方式也有多种。 {43%：主要的分类方式有基于学习策略的分类、按应用领域分类、基于所获取知识的表示形式分类、学习形式分类和综合分类等。} 其中最为普遍接受的是按学习形式分类。

{67%：机器学习按学习形式分类有监督学习（Supervised Learning）、非监督学习（Unsupervised Learning）和强化学习等几种类型。} {70%：实践中应用较多的也就是监督学习和非监督学习。}

{90%：监督学习就是数据样本会告诉计算机在该情形下的正确输出结果，希望计算机能够在面对没有见过的输入样本时也给出正确的输出结果，} 从而达到预测未知的目的。

{89%：根据输出结果是离散还是连续，监督学习可以分为分类问题和回归问题两大类。} {93%：他们在文字、语音、图像识别、垃圾邮件分类与拦截、网页检索、股票预测等方面有着广泛应用。}

{92%：非监督学习是指数据样本中没有给出正确的输出结果信息。} {96%：无监督学习希望从数据中挖掘的正是这一类信息，常见的例子有聚类、关联规则挖掘、离群点检测等等。}

{98%：强化学习是智能系统从环境到行为映射的学习，以使奖励信号（强化信号）函数值最大，强化学习不同于连接主义学习中的监督学习，} {91%：主要表现在强化学习中由环境提供的强化信号是对产生动作的好坏作一种评价（通常为标量信号），} {98%：而不是告诉强化学习系统 RLS（Reinforcement Learning System）如何去产生正确的动作。} {100%：由于外部环境提供的信息很少，RLS必须靠自身的经历进行学习。} {97%：通过这种方式，RLS在行动评价的环境中获得知识，改进行动方案以适应环境。}

2.2最小二乘法回归

2.2.1线性回归模型

考虑到本课题中数据的多维性，这里用多元回归方程表示：

=
0
+
1
1
+
2
2
++

式中： {95%： θ 称为假设模型即映射输入特征 x 与结果 y 的线性函数 h 的参数 (Parameters)，为了简化表示，我们在输入特征中加入 $x_0=1$ ，于是得到：}

=

=0

=

{100%：参数和输入特征 x 都为矢量， n 是输入的特征 x 的个数（不包含 x_0 ）。}

{96%：使得预测值 $h(x)$ 尽可能接近 y ，为了达到这个目的，我们对于每一个参数，定义一个代价函数 (Cost Function) 用来描述 $h(x(i))'$ 与对应的 $y(i)'$ 的接近程度：

=

1

2

=0

(

0

(

()

)

)

2

{100%：于是我们的目标就变为了调整使得代价函数 $J()$ 取得最小值，方法有梯度下降法，最小二乘法等。}

2.2.2 最小二乘法

{97%：最小二乘法是一种直观的直接利用矩阵运算可以得到值的算法。}

{100%：假设函数 f 是将 $m \times n$ 维矩阵映射为一个实数的运算，即}

Function :

并且定义对于矩阵 A ，映射 $f(A)$ 对 A 的梯度为：

=

11

1

1

因此该梯度为 $m \times n$ 的矩阵。

另外，对于矩阵的迹的梯度运算，有如下规则：

=

() =

(

())

=

+

=

(

1

)

{100%：下面，我们将测试集中的输入特征 x 和对应的结果 y 表示成矩阵或者向量的形式，有：}

=

1

2

=

(1)

(2)

()

对于预测模型有：

=

=0

=

,

即：

=

1

,

于是可以很容易得到：

=

1

1

=

1

1

所以可以得到：

1

2

=

1

2

=1

(

(

()

)

)

2

=

{95%：于是，我们就将代价函数 $J()$ 表示为了矩阵的形式，就可以用上述提到的矩阵运算来得到梯度：}

=

1

2

=

1

2

(

+

)

=

1

2

+

=

1

2

2

=

1

2

+

2

=

令上述梯度为0，得到等式：

=

,

于是得到的值：

=(

)

1

。

{100%：这就是最小二乘法得到的假设模型中参数的值。}

2.3 算法实现

2.3.1 Python语言简介

{51%：Python语言诞生于20世纪90年代初，它已被逐渐广泛应用于Web编程和系统管理任务的处理等领域[22]。} {82%：Python语言是一个高层次的结合了编译性、解释性、互动性和面向对象的脚本编程语言。} Python语言具有诸多特点：易于学习、数据库GUI编程、易于阅读、广泛的标准库、易于维护、可扩展性、交互式、可移植、可嵌入等优势。Python语言凭借着自身的诸多优势，广泛应用于系统编程、数据挖掘、图形处理、数学处理、{45%：黑客编程、文本处理、网络编程、数据库编程、Web编程和多媒体应用等领域或方向。}

Python语言区别于其他的编程语言还有一个特点：{64%：具有丰富和强大的库，并能够把用其他语言制作的各种模块联结起来。} 最常见的一种应用情形：将Python快速生成原型程序，然后将其中有特别配置的部分，用更合适的语言编写。比如3D游戏中的图形渲染的模块对于性能要求特别高，这时候就可以用C++重写，然后再将其封装为Python语言可以调用的扩展类库。

{69%：除了自己编写程序模块之外，众多开源的科学计算软件包也提供了Python的调用接口。} {43%：比如著名的计算机视觉库OpenCV、医学图像处理库ITK和三维可视化库VTK。} 而Python专用的科学计算扩展库也有很多。比如经典的科学计算扩展库：SciPy、NumPy和Matplotlib，它们分别为Python提供了数值运算、快速数组处理以及绘图功能。本论文使用这三个标准库，其中更主要的使用了面向机器学习的Scikit-learn标准模块库。

2.3.2 Scikit-learn

{66%：Scikit-learn是一个开源的基于Python语言的科学计算工具包。} {76%：它基于SciPy，针对不同的应用领域已经发展了很多的分支版本，它们统一称为Scikits，就是SciPy工具包的意思。} {95%：在这些分支版本中，最著名的也是专门面向机器学习的一个就是Scikit-learn。} Scikit-learn最早由数据科学家David Cournapeau在2007年提出，它依赖于NumPy和SciPy等程序包的支持，{67%：是Python语言中面向机器学习应用开发的一款开源框架[24]。}

Scikit-learn的基本功能分为六大部分：{92%：分类、回归、聚类、数据降维、模型选择和数据预处理。}

(1) 分类是辨别对象所属的类别，是监督学习的一种。{78%：最常见的应用包括垃圾

邮件识别和图像识别等应用。} 目前Scikit-learn所包含的算法有： {56%：支持向量机（SVM）、最近邻、随机森林、逻辑回归、决策树以及神经网络等等。} 另外由于Scikit-learn本身不支持深度学习，也不支持GPU加速。

{77%：（2）回归是指预测给定对象相关联的连续值的问题，其中最常见的应用场景包括预测药物反应和预测股票价格等问题。} 目前Scikit-learn已经包含的算法有： {41%：支持向量回归（SVR）、弹性网络（Elastic Net）、脊回归（Ridge）、Lasso回归、最小角回归（LARS）和贝叶斯等回归算法。}

{65%：（3）聚类则是指自动识别具有相似属性的对象的问题。} {67%：聚类属于无监督学习的典型，最常见的应用场景主要包括顾客类型细分和试验结果分组等问题。} 目前Scikit-learn已经包含的算法有： {66%：K-均值聚类、均值偏移、谱聚类、分层聚类和DBSCAN聚类算法。}

{73%：（4）数据降维指的是用主成分分析（PCA）、特征选择和非负矩阵分解（NMF）等降维算法来减少需要考虑的随机变量个数，} {58%：其主要应用场景包括效率提升和可视化处理等问题。}

（5）模型选择的主要内容是对于给定的模型和参数的比较、验证和选择。 {79%：模型选择的主要目的是通过参数的调整来提升精度。} 目前Scikit-learn所包含的模块有： {66%：交叉验证、格点搜索和各种预测误差评估的度量函数。}

{57%：（6）数据预处理的主要内容是数据的归一化和特征提取，这是所有的机器学习过程中的首要也是最重要的一个环节。} {89%：归一化是将输入数据转换为具有零均值和单位权方差的变量。} {84%：虽然大多数时候都做不到精确等于零，但是往往会设置一个可接受的范围，一般都要落在0~1之间。} {61%：特征提取所做的是将图像数据和文本数据转换为可用于机器学习的数值变量。}

{83%：总结以上六点来说，Scikit-learn实现了完整用于数据降维、模型选择、特征提取和归一化的完整算法模块，} {73%：同时还提供了每个算法和模块以及其丰富的参考样例和详细的说明文档。}

本文中主要使用了Scikit-learn的回归、模型选择和数据预处理等多个模块。

本文中最小二乘方法主要通过Python、C++等编程语言实现，本文中数据的收集整理主要通过Linux C Shell和Python语言实现。

第三章 28nm标准单元库时序预测

3.1 标准单元库

3.1.1标准单元方法

在半导体设计中，标准单元方法（Standard Cell Methodology）是一种用于专用集成电路（Application Specific Integrated Circuits, ASICs）的数字逻辑的设计方法。 {49%：标准单元方法可以将一个低级的超大规模集成电路（Very-Large-Scale Integration, VLSI）的版图封装成一个抽象的逻辑表示形式，比如与非门。} 这就可以让电路设计师更专注于数字设计，而其他的设计师更专注于物理设计。随着半导体制造业的进步，标准单元方法使得成千上万的逻辑功能相对简单的电路，得以缩放到复杂的百万门级的SOC上。

标准单元是通过晶体管通过组合互联构成具有布尔逻辑函数，比如AND、OR、XOR、XNOR等，存储单元，比如Flip-Flop和Latch。最简单的标准单元比如NAND、NOR、XOR等布尔逻辑功能，尽管更为常用的是一些复杂的布尔逻辑单元：全加器，D触发器等。标准单元的布尔逻辑功能就是它的逻辑视图，其逻辑行为通常表现为真值表、布尔代数方程和状态转换表。其中布尔代数方程通常用于组合逻辑，状态转换表通常用于顺序逻辑。

通常，标准单元的初始设计开发在晶体管级，以晶体管网表或电路的视图的形式。网表是晶体管、晶体管连接、端口与外部电路连接的节点描述。电路图一般都是通过一些计算机辅助设计（Computer aided design，CAD）和电子设计自动化（Electronic design automation，EDA）的程序提供的图形化界面（Graphical User Interface，GUI）来表示。

标准单元除了有数字逻辑的表示，版图（layout）作为最接近实际产品的视图，是由基础层（layer）组成的，基础层对应晶体管的不同结构，布线层（wiring layers）和通过层（via layer）相互连接，然后连接晶体管端口形成的。

最后，布局和线路（PNR）工具可以用来组合的一切，合成的超大规模集成电路的布局，以自动化的方式，从更高层次设计的网表和平面图。

此外，还可以使用许多其他CAD工具来验证单元格视图和模型的其他方面。和其他文件可以创建支持各种工具，利用标准细胞过多的其他原因。{40%：为支持所有标准单元变化而创建的所有这些文件统称为标准单元库。}

3.1.2标准单元库

表3-1标准单元库的分类

标准单元库

组合逻辑单元

Inverter反相器

NAND、NOR 与非、或非门

AND、OR 与、或逻辑

BUFFER 缓冲器

MUX 多路选择器

XOR、XNOR 异或、同或逻辑

AOI、OAI 与或非、或与非

{83%：ADD 加法器包括全加器、半加器}

Clock Cell 包括Clock BUFFER、Clock Inverter、Clock NAND等

时序逻辑单元

Flip-Flop 触发器

LATCH 锁存器

特殊类型单元

FILL cells 填充单元

Tie High、Tie Low cells 电源连接单元

Delay cells 信号延迟单元

二极管单元

Tap cells 阱连接单元

FD CAP cells 去耦合单元

低功耗辅助设计单元

Clock Gating 单元

Power Gating单元

Isolation Cell 隔离单元

Retention Cell 信号保持单元

Level Shift 电平转换单元

Always-on 单元

Well Bias Cell 阱电位偏置单元

{43%：不仅不同工艺尺寸都会有不同的标准单元库，有时相同的工艺尺寸都会有不同的多套标准单元库供用户选择，} 它们在不同的电路密度和速度上有着相应的优势。 {78%：例如中芯国际（SMIC）深亚微米制程下就包括一套高性能的超高速 VHS（Very High Speed）单元库、} {73%：一套密度和速度优化的高速 HS（High Speed）单元库和一套高密度 HD（High Density）单元库。}

{80%：一般而言，标准单元库中包含以下四大类单元：} {44%：组合逻辑单元、时序逻辑单元、特殊类型单元和PMK（Power Management Kit）低功耗辅助设计逻辑单元。}

3.1.3 标准单元库流程

传统标准单元库建立的流程如图：

图3-1标准单元库建立流程

制定设计方案： {80%：在设计设计28nm标准单元之前，我们首先要制定设计方案。} 主要包括： {74%：1）需要设计的单元库器件，2）标准单元库的高度（track数），3）在版图中版面两单元间的中心距距离（Pitch），} {84%：4）MOSFET中 PMOS与

NMOS的宽度比，5) 单元库器件的驱动能力}

单元电路设计： {52%：合理设计出所需要的全部库单元的电路结构和器件尺寸，同时单元的功能和时序也需要通过SPICE等工具进行的相关的验证。}

单元版图设计： 绘制出所有库单元的版图，并使其满足面积最小，同时所有的电路单元保持高度一致。 版图验证主要通过DRC (Design Rule Check) 和LVS (Layout Versus Schematic) 这两步的验证。 其次，还需要根据Foundry所提供的Design Rule进行物理特性的验证。

库模型建立： {78%：一套完整的标准单元库仅包含电路和版图是不够的，还必须包含各种符合国际标准的可被EDA工具识别的库模型文件。} {53%：对于后端设计师而言，使用各种EDA工具，利用Foundry提供的文件，对标准单元库进行仿真、验证。} {76%：这些库文件主要描述了单元的时序信息、版图信息和网表信息三个方面。}

表3-2库模型文件分类

时序信息文件

. lib

{50%：标准单元的Synopsis的时序模型，描述了标准单元的时间、功率特性}

. db

. lib的二进制格式文件

. tlf

{92%：标准单元的TT、FF、SS工艺角的时序库文件}

版图信息文件

. lef

供布局布线使用的版图文件格式，具有工艺信息和各个单元的几何特性的文件，是生成milkyway库所必需的

. gdsii

包含版图信息的文件

网表信息文件

. cdl

标准单元网表，用于LVS检查

. v

Verilog语言格式文件

. vhd

VHDL语言格式文件

符号库文件

. slib

标准单元的符号库文件

库验证： {52%：标准单元库必须通过 Foundry的可制造性验证，同时还要确保标准单元库模型的正确性以及} {89%：应用库模型设计出来的电路在制造出来后时序、功耗、面积多能满足要求。}

Release： {64%：设计并验证标准单元库之后，还需要撰写标准单元库的设计报告、使用手册、数据手册、等相关文档，方便Foundry使用，并将相关资料整理打包。}

3.2 时序

3.2.1 Liberty库

{97%：LIB时序库主要用于描述每个标准单元的基本逻辑功能、延迟时间参数等，其中延迟时间参数是通过投片测试或仿真提取获得的。} {94%：LIB是由 Synopsys公司开发的专门用于描述物理单元的时序以及功耗方面的信息的关键库文件内容，} {41%：一般情况下，LIB文件有库的基本属性和标准单元的具体信息两个部分组成。}

基本属性主要描述了标准单元库的基本信息，主要包含 PVT corner (Process、Voltage、 Temperature) 、文件的版本号、生成日期、电路属性、器件类型、以及时序功耗的查找表等。 {95%：主要用于利用这些单元的功能参数库可以进行电路的Verilog/VHDL数字仿真，确定所设计电路的功能正确与否，检查异步时序电路中的竞争和冒险等工作。}

图3-2LIB文件中功耗数值查询表

LIB文件的主要部分是每个标准单元的功能和有关的延迟参数，主要包括亚阈值电流、输入输出负载以及时序功耗的数值查询表。

如图是UMC 28nm工艺的TT1.05V25C下的标准单元库的LIB文件的INV的功耗查询表。

其中rise_power表示管脚A从0变为1的内部功耗，但不包含开关功耗，开关功耗有电容乘以电压的平方单独运算； index_1表示输入的传输时间； index_2表示输出端电容； values就是当输入输出取不同值时，相对应的内部功耗值。 其中横坐标是index_1，纵坐标是index_2。

图3-3LIB文件中时序数值查询表

如图是UMC 28nm工艺的TT1.05V25C下的标准单元库的LIB文件的INV的时序查询表（部分）。

其中timing就是标准单元的时序描述； cell_fall是下降时间； fall_transition是传输时间； index_1是输入的传输时间； index_2是输出端的电容； values是77的时

序查询表，横坐标是index_1，纵坐标是index_2。

LIB文件包含的时序信息是非常繁多的，详见公式：

{43%: LIB数=阈值电压沟道长度时序模型工艺角电压温度}

以UMC28nmHLP工艺的标准单元库而言，LIB数由以下元素组成。

表3-3LIB文件数的影响因素详表

阈值电压

沟道长度

(纳米)

时序模型

工艺角

电压

(V)

温度

()

RVT

LVT

HVT

30

35

40

NLDM

CCS

ECSM

FF

FFG

TT

SS

SSG

0.855

0.90

0.945

1.05

1.15

-40

0

25

85

125

其中，时序模型分为三种：

NLDM： 非线性延时电压源，适用于40nm以上，LIB文件较小；

CCS： {41%：复合电流源模型，由Synopsys公司开发，适用于40nm以下，时序更为精准，LIB文件较大；}

ECSM： 有效电流源模型，有Cadence公司开发，适用于40nm以下，时序更为精准，LIB文件中等。

{41%：由此可以看出，一套标准单元库会包含数量繁多的LIB文件，大约有1425个LIB文件。}

与此同时，单个LIB文件中具有整个标准单元库的所有电路的时序信息，当阈值电压为HVT，沟道长度是30nm时，就有939个标准单元电路。

由此可见，对于标准单元库而言，

3.2.2 数据收集

由于制造工艺的偏差，同一晶圆的芯片在同一环境下都有不同的工作状态。以MOSFET晶体管为例，尽管他们也有类似的电阻和电容，会有不同的状态，如 FNFP、SNSP、TNTF、FNTP、SNFP，不同的阈值电压等参数。除了制作工艺的偏差，MOSFET晶体管的性能也受到电压和温度的影响。一般来说，仿真的时候会选择不同的温度节点和工作电压节点。

表中有325套Process Corner。

表3-4PVT的选择范围

工艺

FNFP、SNSP、TNTP、FNFP、SNFP

电压 (V)

0.84 0.875 0.91 0.945 0.98 1.015 1.05 1.085 1.12 1.155
1.19 1.225 1.26

温度 ()

-40 -0 25 85 125

本课题选取了十个电路分别是：

INV、AND2、AND3、NOR2B、NOR2、AOI211、XOR3、ADDF、DFFNQ、DFFNSRPQ。

本章仿真均使用基于Bsim4仿真模型的H-Spice仿真工具，Slew设置为1ns。

以D触发器为例，输入端是D，输出端是Q。 路径延时 (Path Delay) 定义为从输入波形的50%到输出波形的90%或者从输入波形的50%到输出波形的10%， 以确保输出波形的电势能稳定达到目标电位。

图说明setup time是时钟达到之前波形需要保持一段时间，hold time是时钟达到之后波形需要保持的一段时间。

以DFFs为例，setup time是从D端到节点m的传输路径时间，hold time有两个值：
{48%：一个是节点m到节点s的传输路径时间，另一个是D端到节点nm的传输路径时间。}
Hold time选取二者的最大值。

图3-4传输路径延时： 从输入波形50%到输出波形10%的传输路径延时

图3-5D触发器的setup time和hold time

图3-6D触发器的逻辑电路图

{41%：本课题中的10个标准单元电路分别有不同数量的路径延迟，D触发器还拥有建立时间和保持时间。}

对于单输入电路而言，以INV为例，当输入端是上升沿时，输出端是下降沿，具有一个路径延时； {43%：当输入端是下降沿时，输出端是上升沿，还具有一个路径延时。}

对于多输入电路而言，当某一输入端翻转，而其他输入端保持电平不变，且输出端也翻转的时候，才具有一个路径延时。

图3-7AND2输入输出波形示意图

以AND2为例，时序包含四种时序：

{46%：当输入A为1时，输入B从0翻转到1时，输出Y从0翻转到1。} 输入B的50%到输出端Y的90%的传输路径延时。

{46%：当输入A为1时，输入B从1翻转到0时，输出Y从1翻转到0。} 输入B的50%到输出端Y的10%的传输路径延时。

{46%：当输入B为1时，输入A从0翻转到1时，输出Y从0翻转到1。} 输入A的50%到输出端Y的90%的传输路径延时。

{46%：当输入B为1时，输入A从1翻转到0时，输出Y从1翻转到0。} 输入A的50%到输出端Y的10%的传输路径延时。

下表是10个电路相对应的传输路径延时的数量：

表3-510个电路的传输路径延时

INV
AND2
AND3
NOR2B
NOR2
AOI211
XOR3
ADDF
DFFNQ
DFFNSRPQ
2
4
6
4
4
16
24
36
3
3

3.3 时序预测

本课题提出了两种方法实现预测标准电路的路径延时： 电路参数方法和时序方法。

3.3.1 电路参数方法

{54%：作为集成电路的最小单元，金属氧化物半导体效应晶体管（MOSFET）的各种特性体现在电路的特性上。} MOSFET根据载流子的极性不同，可以分为NMOSFET和PMOSFET。

MOSFET晶体管都是有漏极（Drain）、栅极（Gate）、源级（Source）与衬底（Body）。
{41%：每个终端都影响着MOSFET的电路特性，不同MOSFET的逻辑组合又具备不同的电路特性。
} {40%：本课题选择了一些MOSFET的电路参数作为机器学习的特性，作为用于机器学习的特征。}

表3-6电路参数表及其含义

I_{vin} (A)

信号上升或下降时 V_{IN} 电压的电流的峰值

I_{vdd} (A)

信号上升或下降时 V_{DD} 电压的电流的峰值

I_{vss} (A)

信号上升或下降时 V_{SS} 电压的电流的峰值

I_{on} (A)

信号上升或下降后MOS管的电流的平均数值

I_{peak} (A)

信号上升或下降后MOS管的电流的峰值

V_{th} (V)

{62%：信号上升或下降时MOS管的阈值电压}

以INV为例，共有325组仿真数据，每组数据有9个电路参数作为拟合的特征（Feature）和一个路径延时作为目标（Target）。表中所示是五组INV的特征和目标值。

表3-7五组INV的电路参数与延时

I_{vin1}

(10-6A)

I_{vdd}

(10-6A)

I_{vss}

(10-6A)

Vth1

(V)

Vth2

(V)

Ion1

(10-9A)

Ion2

(10-9A)

Ipeak1

(10-5A)

Ipeak2

(10-6A)

Delay

(10-11s)

5.868

4.419

7.076

6.802

5.461

1.345

1.020

9.112

1.614

1.422

4.226

3.226

11.97

5.830

3.938

0.5129

0.2629

0.2456

0.4511

0.5768

0.5063

0.3713

0.3713

0.5922

0.4257

4.357

1.663

5.097

2.508

15.85

0.9851

2.142

4.475

0.6069

4.057

1.650

1.064

2.466

1.996

1.358

4.074

3.672

13.06

6.239

3.789

3.146

2.527

1.859

1.456

4.058

{44%：本次学习使用10折交叉验证（10-Fold Cross Validation），每次学习使用32组仿真数据作为训练集，另外的293组仿真数据作为测试集。}

{71%：本次学习选择了拟合优度作为拟合程度的度量，拟合优度（Goodness of Fit）是指回归模型对观测值的拟合程度。} 度量拟合优度的统计量是可决系数（determination coefficients） R^2 。 {93%：可决系数的值越接近1，说明回归直线对观测值的拟合程度越好；} {84%：反之，可决系数的值越小，说明回归直线对观测值的拟合程度越差。}

2

=1

可决系数的表达式如上，其中

是回归平方，

是总平方和。 是表示因变量与所有自变量之间的总体关系。 {82%：拟合优度的可决系数是无量纲系数，有确定的取值范围（0-1），便于对不同资料回归模型的拟合优度进行比较。}

{40%：表3-8十个标准电路延时训练集与测试集的拟合优度}

R^2

INV

AND2

AND3

NOR2B

NOR2

A0I211

XOR3

ADDF

DFFNQ

DFFNSRPQ

Train

0.968

0.973

0.968

0.982

0.989

0.960

0.989

0.995

0.996

0.971

Test

0.962

0.930

0.952

0.973

0.983

0.924

0.962

0.962

0.965

0.940

{50%：本次学习还选择了准确度（Accuracy）作为预测时序准确度的评价标准。} {61%：准确度和相对误差一般由公式表示：}

$=100\%$

$=1$

{50%：其中，是准确度，是实际相对误差，是绝对误差，L为真实值。} 相对误差一般表示为百分数。 {46%：相对误差越接近0，表示预测的时序的准确度越好，预测值与真实值越接近。} {44%：同样的，准确度越接近1，表示预测的时序的准确度越好，预测值与真实值越接近。}

{55%：表3-9十个标准电路延时训练集与测试集的预测准确度}

INV

AND2

AND3

NOR2B

NOR2

AOI211

XOR3

ADDF

DFFNQ

DFFNSRPQ

Train

0.836

0.941

0.968

0.985

0.935

0.887

0.940

0.991

0.993

0.982

Test

0.859

0.796

0.952

0.958

0.771

0.779

0.754

0.954

0.955

0.958

电路参数方法可以取得很好的预测效果，与此同时，所有的电路参数中， I_{vdd} 、 I_{vss} 在电路中只有一个； I_{vin} 的数量对应输入端的数量； V_{th} 、 I_{on} 、 I_{peak} 每一个MOS管都有。对于小电路，电路参数不是很多，然而，对于DFF这种复杂一些的电路，则有三十多个MOS管，{46%：电路参数方法会有大概一百个电路参数作为学习的特征。}

电路参数方法对于标准单元仿真周期，我们采用时间成本和机器资源成本来进行评价。

以UMC28nmHLP为例，通常以50个核来进行仿真，大致需要24小时。采用本方法后，以50个核来进行仿真数据的采集及时序数据预测，大致需要17小时。

3.3.2 时序方法

从Process corner来看，一般而言，时序都随着Process、Voltage、Temperature的变化呈现相对应的变化。

以DFF为例，本方法以其他标准单元电路的时序作为学习的特征，DFF的时序作为目标。其中其他标准单元电路分别是INV、AND2、AND3、NOR2B、NOR2、AOI211、XOR3和ADDF。如图是5个process corner下的DFF的时序，分别是传输路径延时、setup time和hold time。

{43%：本学习也是用10折交叉验证，表列出了DFF的时序拟合优度。} 表列出了DFF的准确度。{82%：可决系数的值越接近1，说明回归直线对观测值的拟合程度越好，同样的，}{48%：准确度越接近1，表示预测的时序的准确度越好，预测值与真实值越接近。}

表3-10DFFs的传输路径延时、setup time和hold time在5个corner下时序

Process corner

DFFNQ

DFFNSRPQ

Path delay

[10-11s]

Setup time

[10-11s]

Hold time

[10-11s]

Path delay

[10-11s]

Setup time

[10-11s]

Hold time

[10-11s]

ttl05v0c

fflp26v125c

ss0p84vn40c

sf1p05v25c

fs1p05v85c

7.766

4.378

72.50

7.999

7.915

8.880

4.760

65.69

9.969

7.918

5.857

3.634

49.26

5.928

6.556

9.717

5.059

95.01

9.793

10.05

11.92

6.276

92.43

12.46

12.24

8.879

5.396

72.91

8.312

10.67

表3-11 DFFs的传输路径延时、setup time和hold time的拟合优度

R²

DFFNQ

DFFNSRPQ

Path delay

Setup time

Hold time

Path delay

Setup time

Hold time

Training set

0.9991

0.9996

0.9989

0.9990

0.9996

0.9990

Testing set

0.9984

0.9987

0.9980

0.9979

0.9992

0.9983

表3-12DFFs的传输路径延时、setup time和hold time的预测准确度

DFFNQ

DFFNSRPQ

Path delay

Setup time

Hold time

Path delay

Setup time

Hold time

Training set

0.9923

0.9907

0.9772

0.9981

0.9946

0.9763

Testing set

0.9912

0.9821

0.9874

0.9901

0.9846

0.9854

从上表可以看出，时序方法可以取得相当好的预测效果。

与电路参数方法相比较，时序方法所需要的特征数目较少，只需要其他电路的时序即可。不需要每个电路的每个MOS管的电路参数。

时序方法对于标准单元仿真周期，我们采用时间成本和机器资源成本来进行评价。

以UMC28nmHLP为例，通常以50个核来进行仿真，大致需要24小时。采用本方法后，以50个核来进行仿真数据的采集以及时序数据预测，大致需要12小时。

3.4本章小结

本章详细阐述了标准单元方法和标准单元库流程，着重讲述了时序库LIB文件，分析了标准单元库仿真的周期时长。并通过电路参数方法和时序方法两种方法，使用最小二乘回归算法预测了标准单元电路的时序，其中电路参数方法的预测时序的准确度能达到94.58%和87.36%；时序方法的预测时序的准确度能达到98.82%和98.68%。并从时间成本和机器资源成本对两种方法进行了评估，电路参数方法节约时间大致有30%，时序方法节约时间大致有50%。

第四章 SRAM编译器

本章以 TSMC28 nm SRAM为例，选择六管单元作为设计方案，介绍了 SRAM编译器，然后详细分析 SRAM编译器的表征（Characterization），{48%：提出预测三种仿真数据的方法，最后对预测的结果和预测方法进行了评价。}

4.1 SRAM基本原理

4.1.1 SRAM架构

TSMC 28nm SRAM架构如图4-1。

图4-1SRAM电路框架图

{51%：SRAM即静态随机存取存储器，SRAM主要由存储单元阵列（Core Cell Array）、地址译码器（Decoder）、灵敏放大器（Sense Amplifier）、控制电路（Control circuit）和缓冲/驱动电路（FFIO）。}

如上图4-1，存储单元阵列有 2^m 行和 2^n 列，分别表示存储阵列有 2^m 根字线（Word line）构成，每个字深存储 2^n 根位线（Bit line）。每个SRAM包含 $2^m 2^n$ 个存储单元。每个存储单元与一根字线和一组反相的位线相连接，并通过它们进行对存储单元的读写操作。

译码器分为行译码器（Row decoder）和列译码器（Column decoder）。行译码器接收地址信号，选中一根字线触发阵列的一行，在进行读操作的时候，字线驱动位线，反相的位线通过灵敏放大器读出存储单元的数据。

通常为了节省功耗和减少数据读取时间，存储器中的字数会远远多于每个字存储的位数，{41%：这样会使得存储器变得长而窄，因为位线过长会增加数据访问的速度。}

4.1.2 SRAM存储单元

图4-2SRAM 6管单元原理图

{44%：图中所示为6管SRAM单元，存储单元使用一根字线（Word Line）、两根反相的位线（bit和bit_b）。}{94%：单元中包括了一对交叉耦合的反相器，并且每根位线链接了一个存取晶体管。}{100%：一对互补的数据存储在交叉耦合的反相器上。}{83%：如果数据受到轻微的干扰，由回路构成的正反馈将使数据恢复到VDD或GND。}{84%：选中这个单元的字线就可以对这个单元进行读出或写入。}

图4-3 TSMC 28nm 6管单元的版图

如图4-3为TSMC 28nm SRAM Compiler的存储单元的版图。

4.1.2SRAM Compiler

一般而言，Memory Compiler属于IPs的一种，目前部分Design House和Foundry都可以向客户提供配套的Memory Compiler。Memory Compiler是设计一套完整的存储器单元模块（SRAM、Flash等）。{94%：设计中包含不同大小的存储器单元模块，尽量做到单元模块的面积和速度最优。}{63%：用户可以通过简单的界面定义实现目标要求的存储器单元模块的GDSII文件和网表文件等。}{85%：由于用户不能更改电路的内部结构和版图，所

以面积和速度就是Memory Compiler的竞争力所在。}

{41%: 常用SRAMs是全定制设计, 全定制设计以存储器密度、速度、功耗和良率为主要设计目标, 全定制设计需要相应较长的设计周期。} 能在较短时间内完成满足设计需求的SRAM, 就意味着在高度竞争的市场中具有强有力的竞争力。 SRAM Compiler就是一种解决方法, SRAM Compiler不仅能节约成本, 而且能在很大程度的缩短设计周期。

{44%: SRAM Compiler是一种能生成不同存储器配置的计算机智能化程序。} SRAM Compiler主要功能包括: {92%: 产生存储器网表 (netlist)、模拟最坏情况下读写路径以产生时序信息、为在原理图中取代 SRAM而} 建立相应的符号、建立带有时序信息的模拟模型、产生对应 SRAM的物理版图、建立电源栅结构等。 {41%: 通常SRAM Compiler用来为诸如门阵列或标准单元电路的专用集成电路 (ASICs) 生成存储器。}

4.1.2.1 SRAM Compiler设计流程

图4-4 SRAM Compiler设计流程图

{85%: SRAM Compiler通过调用设计电路结构中的子模块的版图来产生存储器模块版图, 所以不同的电路设计结构得到的版图会对SRAM Compiler的面积有很大的影响。}

如图4-4是SRAM Compiler的设计流程, 发开一套SRAM Compiler的主要流程如下: 根据客户的需求以及市场信息, 提出对 SRAM的设计需求, 主要是 SRAM所具备的一些功能和特征, {40%: 并对存储器单元进行评估, 并检查是否满足存储器设计的速度和面积要求;} 然后对SPEC的配置进行电路图设计及预仿真, 其次是页元 (Leaf cell) 的版图设计, 通过DRC和LVS规则对电路图和版图进行优化调整; 使用 SRAM Compiler的 Tiling Engine来产生存储器模板 (Memory Instance) 版图, 最后是对不同配置的 Instance进行表征 (Characterization), 主要通过时序 (Timing)、功耗 (Power) 和电容 (Capacitance) 来体现。

在Memory Characterization环节中, 主要是仿真工作和仿真数据收集。 实际设计中, 仿真工作的时间几乎占据了全部的Memory Characterization环节。 因为仿真工作的数量由SPEC配置和PVT共同决定。 而SPEC与PVT参数包含了特征、MUX、Bank、Word、Bit、Input Slew、Output loading、Process Corner、Voltage以及Temperature等。 参数的繁多决定了仿真工作的数量的繁多。

4.1.2.2 SRAM Compiler整合流程

图4-5 SRAM Compiler整合流程图

Leaf cell的版图和电路在通过 DRC和 LVS之后, 通过 Tiling Engine的脚本生成不同 instance以及 instance的版图文件 (LEF、 GDSII文件) 和电路描述文件 (Verilog语言文件、 SPICE文件) ;

Characterization环节通过设计者给出的 Instance列表对 instance抽象成 CP (Critical Path), 通过 Word、 Bit, Feature的变化来改变 CP中的各个参数, 模拟对应 Instance的行为, 节省了仿真时间, 同时还不是准确性。 针对不同instance的SPEC配置文件的变量参数进行配置, 通过脚本自动生成仿真文件, 并进行仿真;

仿真完成后, 通过测量描述语言对仿真结果文件进行数据抽取, 生成Rawdata文件, 并通过计算文件生成最后的LIB文件。

{42%：最后将 LIB文件、 LEF文件、 GDSII文件、 Verilog文件和 SPICE文件整合，编写 Compiler实现程序，使得用户可以} 通过自定义 instance的大小与功能自动生成 SRAM及其网表文件、版图文件与表征数据库等。

{58%：通过 Compiler工具中的各种引擎模块调用设计中相应的设计输入文件，如SPEC配置文件、 Compiler的物理单元库（ GDS），} 网表单元库（ Netlist）、Memory Development Language（ MDL）语言脚本等文件，具体需要的文件见图；Memory Compiler的输出文件结构如图，大致有GDSII、LEF、SPICE和Verilog文件等。

图4-6Memory Compiler输入输出文件结构

4.2 Memory Characterization

4.2.1仿真工作

Memory Characterization是SRAM Compiler设计中非常重要的环节，主要是instance的仿真工作以及仿真数据的收集。其中仿真数据主要是时序和功耗。

Characterization的仿真工作具有一定的重复性，可以通过脚本的实现来生成Test bench，进行仿真、数据检查以及数据收集。Characterization减少了设计者的仿真时间，同时也可以减少一些人为的错误。

仿真的工作量主要由SPEC配置和PVT共同决定，SPEC配置有FEATURE、MUX、BANK、WORD、BIT、INPUT SLEW和OUTPUT LOADING等。

4.2.2 特征

本课题SRAM Compiler所采用的工艺是TSMC28nm的HPC+工艺版本，本课题SRAM Compiler所采用的功能如下：

Synchronous read and write operations
Column Mux options for the best aspect ratio
Fully customized layout density available in 0.9V 10%
Three power management modes: Light Sleep, Deep Sleep and Shut Down to reduce static power
Bit write and global write operations available
Extra margin control option
Special test modes externally bypass read and write self-timing circuits
Redundancy inside to repair defective cell

表4-1TSMC28nm SRAM Compiler具备的特征

具有deep sleep模式

2

具有shut down模式

P

具有light sleep功能

B

具有Bit write功能

I

具有IO redundancy功能

M

具有BIST功能

E

具有HVT Periphery功能

L

具有LVT Periphery功能

用户可以通过定义不同的word、bit和mux来优化存储器，从而实现面积、速度和功耗的最佳平衡。以上的特征都被集成到Memory Compiler中，用来实现系统应用和用户需求。在设定一些参数之后，Compiler就可以在很短的时间内生成memory instance的data sheet、GDSII Layout、LVS netlist和PR model等。

通常，Memory Compiler所提供的全部功能，用户在自定义生成SRAM的时候，根据需求选择其中几个或者全部的功能。这就要求在进行仿真的时候，针对不同的特征组合进行不同的仿真，这也是仿真工作量繁多的重要原因。

以下为需要进行仿真的特征组合：

—

1, 2, b, i, m, p

12, 1b, 1i, 1m, 1p, 2b, 2i, 2m, 2p, bi, bm, bp, im, ip, mp

12b, 12i, 12m, 12p, 1bi, 1bm, 1bp, 1im, 1ip, 1mp,

2bi, 2bm, 2bp, 2im, 2ip, 2mp, bim, bip, bmp, imp

12bi, 12bm, 12bp, 12im, 12ip, 12mp, 1bim, 1bip,

1bimp, 1imp, 2bim, 2bip, 2bmp, 2imp, bimp

12bim, 12bip, 12bmp, 12imp, 1bimp, 2bimp

12bimp

共计64组特征。 针对每一组特征都有相应的仿真工作需要完成。

4.2.3 INSTANCE

用户可以通过定义不同的word、bit、mux和bank来定义需求的存储器。

{43%：通常随着存储阵列容量的变大，那么字线和位线就会变得更长。} {54%：增加字线和位线的长度会使电容变大，从而导致延时和功耗的增加。} 因此，通常会将大容量的存储阵列分割成为多个较小的存储阵列，这样的较小的存储阵列就被称为体（Bank）。

{53%：通常为了节省功耗和减少数据读取时间，存储阵列中的字数会远远多于每个字存储的位数，} {41%：这样会使得存储器变得长而窄，因为位线过长会增加数据访问的速度。}

图4-7Mux示意图

通常采用的另一个做法就是减少位线的长度。 如图将一根正常的位线分割成两部分，每根位线的长度与其所驱动的单元数都减半，从而减少了位线的负载。 这样子分割位线的做法叫做Column Mux。 其中每根位线的长度被分成几份，Mux就等于几，且Mux总是2的幂值。 在具体的SRAM Compiler设计中，会提供很多不同的字线、位线和Mux来供用户进行选择。

本课题的SRAM Compiler是小面积的存储器，设计中不分Bank，因此Bank的值为1。

Mux根据SPEC文件定义为2和4。 当Mux取2时，Word最小取32，最大取512，步长为4，Bit最小取8，最大取144，步长为2； 当Mux取4时，Word最小取64，最大取1024，步长为8，Bit最小取8，最大取72，步长为1。

表4-2字长根据Mux不同的选择范围

Word	Range
Column Mux=2	
32, 36, 40, 508, 512	
Column Mux=4	
64, 72, 80, 1016, 1024	

表4-3位线根据Mux不同的选择范围

Bit	Range
Column Mux=2	

8, 10, 12, 142, 144

Column Mux=4

8, 9, 10, 71, 72

当Mux选定2或者4的时候，选择不同的Word和Bit的值即可确定Memory Instance的大小。从上表可以看出，当Mux=2的时候，Word可取120个值，Bit可以取140个值；当Mux=4的时候，Word可以取120个值，Bit可以取64个值。由此可计算出本课题的SRAM Compiler的Instance个数共有24480个。理论上，用户自定义生成的Memory Instance的可以是24480中的任何一种，这就要求每一个Memory Instance都有相应的仿真数据，供用户查询。仿真阶段完成所有的Instance的仿真是不可能实现的，通常设计公司会提供比较典型的几个Instance的仿真数据。而客户需求所提供的Instance之外的仿真数据的时候，就需要设计公司重新进行仿真。

4.2.4 IO

在IO部分中，电路的端口主要有A、CK和D0。输入端口和时钟端口的slew的不同对电路的表征有不同的影响，在输出端口的负载的大小对电路的表征也有不同的影响。

在SPEC文件中，input slew和output loading都设置七个值。其中A作为数据输入的端口，采用input slew的值，称之为Data slew，CK作为时钟，也采用input slew的值，称之为CK slew，D0作为输出端，选有output loading的值，称之为load。

本课题中，input slew和output loading的值如下表：

表4-4：input slew和output loading数值表

Input slew

Data slew

0.004, 0.050, 0.100, 0.200, 0.500, 0.750, 1.000

CK slew

0.004, 0.050, 0.100, 0.200, 0.500, 0.750, 1.000

Output loading

Load

0.001, 0.020, 0.050, 0.100, 0.250, 0.300, 0.350

对于instance而言，Data slew、CK slew和Load选择不同的数值的时候，都有不同的电路表征。也就是说对于某个确定的Instance而言，在IO中就有343种状态。

4.2.5 PVT Corners

{87%：从晶片中提取相对应的器件模型时，片上 NMOS和 PMOS的测试结构显示出不同

的门延时，} {100%：而这些角的实际选取是为了得到可接受的成品率。} {100%：各种工艺角和极限温度条件下对电路进行仿真是决定成品率的基础。}

{65%：在不同批次和不同的晶圆之间，MOSFETs的性能变化很大。} {100%：为了在一定程度上减轻电路设计任务的困难，工艺工程师们要保证器件的性能在某个范围内，} {100%：大体上，他们以报废超出这个性能范围的芯片的措施来严格控制预期的参数变化。}

{96%：通常提供给设计师的性能范围只适用于数字电路并以工艺角（Process Corner）的形式给出。}

图4-8PVT Corner的示意图

如图，其思想是： {100%：把NMOS和PMOS晶体管的速度波动范围限制在由四个角所确定的矩形内。} 这四个角分别是：快速NMOSFET和快速PMOSFET，慢速NMOSFET和慢速PMOSFET，快速NMOSFET和慢速PMOSFET，慢速NMOSFET和快速PMOSFET。 {90%：例如具有较薄的栅氧、较低阈值电压的晶体管，就落在快速角的附近。}

{87%：从载流子迁移率（Carrier mobility）的角度而言，载流子迁移率是指在载流子在单位电场作用下的平均漂移速度。} 工艺角的命名方式是两个字母组合，前一个字母表示NMOS的工艺角，后一个字母表示PMOS的工艺角。 工艺角的类型通常有：快速（Fast）、慢速（Slow）和正常（Typical）。 以TT为例，就是指正常速度的NMOS管和正常速度的PMOS管。

{87%：单一器件载流子迁移率的结果是呈正态分布的，均值在TT，最小最大限制值为SS与FF。} {56%：从NMOS和PMOS管的迁移率分布图而言，这3种工艺角覆盖大约3 sigma即约99.73% 的范围。} {100%：对于工艺偏差的情况有很多，比如掺杂浓度，制造时的温度控制，刻蚀程度等，} {100%：所以造成同一个晶圆上不同区域的情况不同，以及不同晶圆之间不同情况的发生。} {100%：这种随机性的发生，只有通过统计学的方法才能评估覆盖范围的合理性。}

本课题采用3 Corner model，他们是TNTP，SNSP和FNFP。

{90%：设计除了要满足上述的Process Corner外，还需要满足电压与温度等条件，形成的组合称为PVT（process， voltage， temperature）。}

表4-5： PVT配置表

Process

TT、SS、FF

Voltage/V

0.72、0.81、0.9、0.99、1.08

Temperature/

-40、25、125

{53%：如表，电压设定数值分别是中位电压的80%、90%、100%、110%和120%。} {93%：设计时还需要找到最好最坏情况，时序分析中将最好的条件（Best Case）定义为速度最快的情

况，而最坏的条件(Worst Case)则相反。}

{66%：根据不同的需要，会有不同的PVT组合，以下是一些标准的分析条件：}

WCS (Worst Case Slow)：slow process、high temperature、lowest voltage

TYP (Typical)：typical process、nominal temperature、nominal voltage

WCL (Worst Case Cold)：slow process、lowest temperature、lowest voltage

BCF (Best Case Fast)：fast process、lowest temperature、high voltage

ML (Maximal Leakage)：fast process、high temperature、high voltage

TL (Typical Leakage)：typical process、high temperature、nominal voltage

前四个的PVT组合主要用于时序分析，后两者PVT组合通常用于功耗分析。与前面说设计师通常只会提供一部分Instance的仿真数据一样，设计师通常也只会提供小部分PVT的仿真数据。当用户需求超出了最初的设计范围时，需要重新进行仿真工作。

4.3 本章小结

本章主要介绍了SRAM的基本框架和基本电路结构，然后描述了SRAM Compiler的设计流程与主要文件类型；着重对Compiler的表征环节进行了阐述，主要分析了Instance生成所受的因素影响。并详细对特征、word、bit、mux、bank、input slew、output loading和PVT corners进行详细解释，间接地表达了memory instance数量与仿真工作量的繁多。

第五章表征数据预测

5.1时序

5.1.1 时序数据

5.1.1.1 读操作时序

SRAM有三种工作状态：保持 (Standby)、写入 (Write) 和读出 (Read)。

SRAM的数据保持时，芯片使能信号CEB处于无效状态，无论其他信号是什么状态，都不会对存储单元进行任何操作。

图5-1：读操作波形示意图

{40%：SRAM读操作时序如图，其中WEB为写使能，CK是时钟信号，CEB为片选中信号，A为地址信号，D0为数据输出信号。}

表5-1读操作时序

Parameter

Symbol

Read cycle time

Trc

Data access time at the CK rising edge

Tcq

Output data hold time after the CK rising edge

Toh

Address setup time before the CK rising edge

Tas

Address hold time after the CK rising edge

Tah

GWEB, WEB setup time after the CK rising edge

Tws

GWEB, WEB hold time after the CK rising edge

Twh

CEB setup time before the CK rising edge

Tces

CEB setup time after the CK rising edge

Tceh

Clock high/low pulse widths

Thpw/Tlpw

{46%: SRAM读取数据的时候, 芯片使能信号 CEB处于低电平(有效状态), 写使能信号处于高电平(无效状态), } {43%: 地址信号 A在时钟信号 CK的上升沿到来的前后都保持一段时间, 输出信号 DO在延时之后会得到相应的数据读出。} 其中在时钟信号上升沿到来的前后都需保持一段时间的信号分别有地址信号A、写操作使能信号和芯片使能信号CEB。 {50%: 这三个信号在时钟信号上升沿到来之前保持的时间就是建立时间(Setup time), 比如: } Tas、Tws和Tces。 {46%: 相应的在时钟信号上升沿到来之后信号保持

的时间就是保持时间 (Hold time)，比如Tah、Twh和Tceh。} 除此之外时钟信号CK处于高电平的时间为Thpw，处于低电平的时间为Tlpw，时钟信号的周期为Trc。 输出信号D0从时钟信号上升沿到输出信号开始翻转的时间为Toh，而从时钟信号上升沿到输出信号翻转结束的时间为Tcq。 其中Tcq越小说明SRAM工作越快。

对于SRAM的延时Tcq，它主要分布在三个方面： {100%：首先是译码延时，其次是存储阵列上的延时，另外还有时钟电路、放大电路、输出驱动等延时。}

$T_{cq} = T_{cq}$

cksr

+Tcq

instance

+Tcq(load)

由公式可以看出，Tcq主要有三部分组成： CKsr的延时、阵列的延时和读出路径的延时，其中cksr是Clock slew rate, load是output loading。

图5-2Tcq组合示意图

其中tcq(cksr)只取最大instance和最小loading时的值； tcq(inst)只取最小cksr和最小loading时的值，tcq(load)取最小cksr和最大inst的值。

因为CK Slew和output loading都有7个数值，所以每一个instance的每一个时序都有77的查询表。

5.1.1.2 写操作时序

图5-3写操作波形示意图

表5-2写操作时序表

Parameter

Symbol

Write cycle time

Twc

Address setup time before the CK rising edge

Tas

Address hold time after the CK rising edge

Tah

GWEB, WEB setup time after the CK rising edge

Tws

GWEB, WEB hold time after the CK rising edge

Twh

Input data setup time before the CK rising edge

Tds

Input data hold time after the CK rising edge

Tdh

CEB setup time before the CK rising edge

Tces

CEB setup time after the CK rising edge

Tceh

Output data invalid after the CK rising edge

Twdx

Output data valid after the CK rising edge

Twdv

Clock high/low pulse widths

Thpw/Tlpw

{45%: SRAM写操作时序如图，其中 WEB为写使能， CK是时钟信号， CEB为片选中信号，} {58%: A为地址信号， DI为数据输入信号， DO为数据输出信号。}

{49%: SRAM写数据的时候，芯片使能信号 CEB处于低电平（有效状态），写使能信号处于低电平（有效状态），} {43%: 地址信号 A在时钟信号 CK的上升沿到来的前后都保持一段时间，输出信号 DO在延时之后会得到相应的数据读出。} {41%: 其中在时钟信号上升沿到来的前后都需保持一段时间的信号分别有地址信号A、写操作使能信号、数据输入信号DI和芯片使能信号CEB。} {50%: 这四个信号在时钟信号上升沿到来之前保持的时间就是建立时间（Setup time），比如：} Tas、Tws、Tds和Tces。 {46%: 相应的在时钟信号上升沿到来之后信号保持的时间就是保持时间（Hold time），比如Tah、Twh、Tdh和Tceh。} 除此之外时钟信号CK处于高电平的时间为Thpw，处于低电平的时间为Tlpw，时钟信号的周期为Twc。 输出信号DO从时钟信号上升沿到输出信号开始翻转的时间为Tw dx，而从时钟信号上升沿到输出信号翻转结束的时间为Twdv。

5.1.1.3 电源模式时序

图5-4Light sleep模式下电路各端口的波形示意图

图5-5Deep sleep模式下电路各端口的波形示意图

图5-6Shut down模式下电路各端口的波形示意图

Parameter

Symbol

LS fall setup time before CK rises

Tflss

LS rise setup time before CK rises

Trlsh

DS fall setup time before CK rises

Tfdss

DS rise setup time before CK rises

Trdsh

DS rise to output hold previous data time

Tdsoh

DS rise to all output fall to logic 0 delay

Tdsq

SD fall setup time before CK rises

Tfsds

SD rise setup time before CK rises

Trsdh

SD rise to output hold previous data times

Tsdoh

SD rise to all output fall to logic 0 delay

Tsdq

表5-3电源模式时序表

在Light sleep模式，SRAM可以在不到一个完整的时钟周期之内恢复（wake up），但其静态功耗也是三种模式中最大的；Shut down模式具有最少的静态功耗，但其恢复时间是三种模式中最长的；Deep sleep模式的静态功耗和恢复时间都是介于两者之间。

以 W32 B8 M2 B1为例,当 ck slew=0.004, load=0.001, data slew=0.004时, Tcq在45个 corners下的值。

图5-7全corners组合下Tcq的时序柱状图

本课题分别从读操作时序、写操作时序和电源模式中选择20个时序, 分别是 Tcq、Toh、Troslp、Tfoslp、Tsdq、Tsdoh、Tdsq、Tdsoh、Tas、Tah、Tds、Tdh、Tceh、Tces、Tws、Twh、Thpw、Tlpw、Trc、Twc。

5.1.2 时序预测

5.1.2.1 相同Instance预测

本方法可以实现通过某一确定Instance的CK Slew、DATA Slew和LOAD的时序预测其他所有的CKSR、DATASR和LOAD组合的时序。

本方法使用 CK slew=0.004, Data slew=0.004, Load=0.001 (组合1) 的时序, 去预测了其他不同 CK slew、Data slew和 Load组合的时序。目标组合一共有7组, 分别是:

组合2: CK slew=0.004, load=0.001, data slew=0.500;

组合3: CK slew=0.004, load=0.500, data slew=0.004;

组合4: CK slew=0.004, load=0.500, data slew=0.500;

组合5: CK slew=0.500, load=0.001, data slew=0.004;

组合6: CK slew=0.500, load=0.001, data slew=0.500;

组合7: CK slew=0.500, load=0.500, data slew=0.004;

组合8: CK slew=0.500, load=0.500, data slew=0.500。

本次预测采用8组corners作为训练集: FG1 P08 VN40 C、FG0 P9 V25 C、FG1 P08 V25 C、TT0 P9 VN40 C、TT0 P9 V125 C、SG0 P81 VN40 C、SG0 P99 VN40 C和 SG0 P9 V125 C。其他的37组Corners作为测试集。

以 CK slew=0.500, load=0.500, data slew=0.500为目标组合, 其测试集的预测时序与仿真真实时序的对比图如图。

图5-8CK slew=0.500, load=0.500, data slew=0.500时的

{41%: 测试集corners的Tcq的仿真真实值与预测值及其预测准确度}

从图中可以看出37组Corners下Tcq的仿真值和预测值相差很小, 可以从预测的准确度来观察仿真值和预测值的差异。

例中对测试集的37组数据的准确度求平均, 得到本组Tcq预测的平均准确度为99.83%。

图5-9测试集corners的全时序仿真数据的预测平均准确度

上图为目标组合的测试集所有时序的平均准确度。 可以从图中观察到七个目标组合的测试集在37组 PVT Corners下的时序的平均准确度都处于97%以上， {45%：其中组合3和组合5的 Tds的预测准确度约为97%，除此之外所有的平均准确度都达到98%以上，} 而且大部分的时序的预测平均准确度可以达到99%。 由此可以得出结论： 本方法在 TSMC28 nm工艺下的同一 Instance，通过某一确定的 Input slew和 output loading的时序仿真数据， 实现其他 Input slew和 output loading组合的时序仿真数据的预测，可以预测得到准确度很高的时序仿真数据， 同时可以很有成效的减少仿真工作量，节省仿真时间和仿真资源。

5.1.2.2 不同Instance预测

本课题中选用了4个 instance，他们分别是 W32 B8 M2 B1、 W32 B144 M2 B1、 W512 B8 M2 B1和 W512 B144 M2 B1。 Instance的形状与阵列的大小密切相关，从版图中我们可以看出存储阵列几乎占据了SRAM版图98%的面积。4.1中讲到通常阵列会被设计成行数较少、列数较多的结构， {40%：因为位线的负载是通过并联的，过长的位线会使得 SRAM读写的速度变得很慢。} {47%：相同的存储容量的SRAM会因为其字线和位线的不同具有不同的读写速度，也就是时序。}

Instance的形状与字数和位数线性相关，从版图上看Instance总是矩形，其宽通常为Word/Mux，其长通常为Bit/Mux。

表5-44个Instance的特征、存储容量

Inst
W32B8M2B1
W32B144M2B1
W512B8M2B1
W512B144M2B1
WordBit
1616
16288
25616
256288
特征
面积最小
矮扁
高瘦

面积最大

存储容量

0.4MB

4.5MB

4MB

72MB

本时序预测方法通过某个Instance的全corner来预测另一个instance的时序。可以实现相同CK slew、Data slew、Load的时候，通过单个instance的全corner时序以及目标instance的6、8组corners的时序，可以实现预测其他corners的时序。

方法一：

通过 W32 B8 M2 B1和 W32 B144 M2 B1的8组 corners时序数据作为训练集，其中 W32 B8 M2 B1的8组 corners时序数据作为训练集的特征（Feature），而 W32 B144 M2 B1的8个 corners时序数据作为训练集的目标（Target）。W32 B8 M2 B1和 W32 B144 M2 B1的剩余37组时序数据作为测试集，{43%：其中 W32 B8 M2 B1的37组时序数据作为测试集的特征，W32 B144 M2 B1的37组时序数据作为测试集的目标。} 使用最小二乘法对训练集特征和目标进行拟合，然后测试集的特征作为输入，得到测试集的目标预测值，并与测试集的目标真实值进行对比。

图5-10W32B144M2B1的测试集目标Tcq的仿真值、预测值以及预测准确度

本方法中的训练集的8组 Corners分别是 FG1 P08 VN40 C、FG0 P9 V25 C、FG1 P08 V25 C、TT0 P81 V25 C、TT0 P99 V25 C、SG0 P72 VN40 C、SG0 P9 V25 C和 SG1 P08 V125 C。

如图是本次预测的W32B144M2B1的测试集的Tcq的仿真值、预测值以及预测准确度。

{44%：对39组Tcq的预测准确度进行求平均值，得到本次学习的平均预测准确度为99.69%。}

{47%：从表中也可以看出所有Tcq的预测准确度均处于99%以上，预测效果很好。}

{43%：图5-11三个目标Instances测试集的全时序预测平均准确度}

上图为 W32 B144 M2 B1、W512 B8 M2 B1和 W512 B144 M2 B1的测试集的所有时序的平均准确度折线图。从折线图中可以观察到三组 Instance的测试集在37组 PVT Corners下的时序的平均准确度都处于97%以上，其中 W32 B144 M2 B1和 W512 B144 M2 B1的 Tas的预测平均准确度约为97%，{41%：W512 B8 M2 B1的 Tds的预测平均准确度约为97%，除此之外所有的平均准确度都达到98.5%以上，} 而且大部分的时序的预测平均准确度可以达到99%。

方法二：

本方法通过选择 W32 B8 M2 B1和 W32 B144 M2 B1的6组 corners时序数据作为训练集，其中 W32 B8 M2 B1的6组 corners时序数据作为训练集的特征，而 W32 B144 M2 B1的6个 corners时序数据作为训练集的目标。W32

B8 M2 B1和 W32 B144 M2 B1的剩余39组时序数据作为测试集， {43%：其中 W32 B8 M2 B1的39组时序数据作为测试集的特征， W32 B144 M2 B1的39组时序数据作为测试集的目标。} 使用最小二乘法对训练集特征和目标进行拟合，然后测试集的特征作为输入，得到测试集的目标预测值，并与测试集的目标真实值进行对比。

本方法中的训练集的6组 Corners分别是 FG1 P08 VN40 C、 FG1 P08 V25 C、 TT0 P81 V25 C、 TT0 P99 V25 C、 SG0 P72 VN40 C和 SG0 P72 VN40 C。

图5-12W32B144M2B1的测试集目标Tcq的仿真值、预测值以及预测准确度

如图是本次预测的W32B144M2B1的测试集的Tcq的仿真值、预测值以及预测准确度。
{43%：对39组Tcq的预测准确度进行求平均值，得到本次学习的平均预测准确度为99.68%。}
{47%：从表中也可以看出所有Tcq的预测准确度均处于99%以上，预测效果很好。}

图5-13三个目标Instances测试集的全时序预测平均准确度

上图为 W32 B144 M2 B1、 W512 B8 M2 B1和 W512 B144 M2 B1的测试集的所有时序的平均准确度折线图。从折线图中可以观察到三组 Instance的测试集在39组 PVT Corners下的时序的平均准确度都处于95%以上，其中 W32 B144 M2 B1和 W512 B144 M2 B1的 Tas、 Tws的预测平均准确度约为97%， {41%：W512 B8 M2 B1的 Tds的预测平均准确度约为95%，除此之外所有的平均准确度都达到98.5%以上，} 而且大部分的时序的预测平均准确度可以达到99%。

方法三：

通过 W32 B8 M2 B1和 W32 B144 M2 B1的8组 corners时序数据作为训练集，其中 W32 B8 M2 B1的8组 corners时序数据作为训练集的特征，而 W32 B144 M2 B1的8个 corners时序数据作为训练集的目标。W32 B8 M2 B1和 W32 B144 M2 B1的剩余37组时序数据作为测试集， {43%：其中 W32 B8 M2 B1的37组时序数据作为测试集的特征， W32 B144 M2 B1的37组时序数据作为测试集的目标。} 使用最小二乘法对训练集特征和目标进行拟合，然后测试集的特征作为输入，得到测试集的目标预测值，并与测试集的目标真实值进行对比。

本方法中的训练集的8组 Corners分别是 FG0 P81 VN40 C、 FG0 P99 VN40 C、 FG0 P9 V25 C、 TT0 P9 VN40 C、 TT0 P9 V125 C、 SG0 P81 VN40 C、 SG0 P99 VN40 C和 SG0 P9 V125 C。

图5-14W32B144M2B1的测试集目标Tcq的仿真值、预测值以及预测准确度

如图是本次预测的W32B144M2B1的测试集的Tcq的仿真值、预测值以及预测准确度。
{50%：对37组Tcq的预测准确度进行求平均值，得到本次学习的平均预测准确度为99.64%。}
{45%：从表中也可以看出所有Tcq的预测准确度均处于98.8%以上，预测效果很好。}

图5-15三个目标Instances测试集的全时序预测平均准确度

上图为 W32 B144 M2 B1、 W512 B8 M2 B1和 W512 B144 M2 B1的测试集的所有时序的平均准确度折线图。从折线图中可以观察到三组 Instance的测试集在37组 PVT Corners下的时序的平均准确度都处于96%以上，其中 W32 B144 M2 B1和 W512 B144 M2 B1的 Tas的预测平均准确度约为97.5%， {42%：

W512 B8 M2 B1的 Tds的预测平均准确度约为96%，除此之外所有的平均准确度都达到98.5%以上，} 而且大部分的时序的预测平均准确度可以达到99%。

方法比较

在以上三个方法中，都是通过某一确定 Input slew和 output loading的 Instance的时序仿真数据，以及目标 Instance的部分 corners的时序仿真数据，{41%：通过最小二乘法进行回归训练，并通过训练模型进行测试集的预测。}

在以上三个方法中，方法一是用8个 corners进行训练集范围内部预测，方法二使用6个 corners进行训练集范围内部预测，方法三通过8个 corners进行全 corners范围预测。{41%：三种方法的Tcq预测准确度的最小值分别是99.05%、99.05%、98.77。} 三种方法的所有时序的预测平均准确度的最小值分别是96.99%、95.31%、96.24%。

从 Tcq的准确度和全时序的平均准确度来看，方法一适用于有较多 corners仿真数据的目标 Instance，且已有 corners是处于 PVT corners的边缘，并对预测时序数据有非常高准确度和精确度的情况；方法二适用于有较少 corners仿真数据的目标 Instance，且已有 corners是处于 PVT corners的边缘，{42%：并对预测时序数据有比较高的准确度和精确度的情况；} 方法三适用于有较多 corners仿真数据的目标 Instance，且已有 corners不是处于 PVT corners的边缘，{44%：并对预测时序数据有非常高准确度和精确度的情况。}

由此可以得出结论：本方法在 TSMC28 nm工艺下某一确定的 Input slew和 output loading的某一 Instance的时序仿真数据，{45%：实现其他 Instance的时序仿真数据的预测，提供了三种使用不同情况的预测方法，} {42%：且三种方法都可以得到准确度很高的时序仿真数据。}

以本套SRAM Compiler为例，以50个Intel (R) Xeon 频率为3.60MHZ的4核CPU，其全corners时序表征仿真工作量需要大致340天。在本时序预测方法应用于仿真工作中，本套SRAM Compiler全部仿真数据获得大概为120小时。缩减了约65%的仿真时间。由此可见，本时序预测方法可以很有成效的减少仿真工作量，同时节省仿真时间和仿真资源。

5.2功率

5.2.1 功率数据

图5-16CK、A、DI的波形示意图

如图，读电流的测量时间从 CK的下降沿到下一个下降沿，在读周期中只有 DI、A、CK在翻转，其他的 Pin都是 standby，且 A和 DI应该变化两次。

写电流的测量时间从 CK的下降沿到 CK的下一个下降沿，在写周期中只有 DI、A、CK在翻转，其他的 Pin都处于 standby，且 A和 DI都翻转两次。

在 light sleep mode、deep sleep mode和 shut down mode下，所有的 Pin都保持原值，然后在分别在 light sleep、deep sleep和 shut down的最后几个周期，选择两个周期测量。

在功耗中主要需要采集的功率数据主要有：

dc: standby状态下的电流的平均值

ac_read: 读操作电流平均值

ac_write: 写操作电流平均值

ls: light sleep模式下电流平均值

ds: deep sleep模式下电流平均值

sd: shut down模式下电流平均值

通常为了节省功耗和减少数据读取时间，存储器中的字数会远远多于每个字存储的位数，{41%：这样会使得存储器变得长而窄，因为位线过长会增加数据访问的速度。} 字深的长度和位长的大小对功率也有同样的影响。 本方法考虑到样本的多样性，以及数据的差异性，采用4个形状各异的instance作为训练集，如图上上： W32 B8 M2 B1、 W32 B144 M2 B1、 W 512 B8 M2 B1和 W512 B144 M2 B1分别是字长 的 {61%：极大值和极小值、位深的极大值和极小值的组合，} 也就代表了不同字长、位深对于功耗大小响应的极致体现。

本方法共采集了18个instance的全corner仿真数据，通过程序处理和运算得到每组仿真数据分别为6个参数。 共得到18组456的数据集。 鉴于功率参数的复杂性，本方法的训练集采用4个 instance的数据集， 以其他14个 instance的数据作为测试集，分别进行了14次回归学习和预测。 分别是 W80 B8 M2 B1、 W128 B8 M2 B1、 W192 B8 M2 B1、 W256 B8 M2 B1、 W320 B8 M2 B1、 W384 B8 M2 B1、 W448 B8 M2 B1、 W80 B144 M2 B1、 W128 B144 M2 B1、 W192 B144 M2 B1、 W256 B144 M2 B1、 W320 B144 M2 B1、 W384 B144 M2 B1、 W448 B144 M2 B1等14个 Instances。

因为功率参数受驱动电压波动的影响会有很明显的波动，通常在仿真阶段 SRAM Compiler会选择三种电压作为仿真基准电压， 分别是 HVT (High Voltage Threshold)、 SVT (Standard Voltage Threshold) 和 LVT (Low Voltage Threshold) 。 {44%：通常功耗会随着阈值电压的增加而减少，就是相同条件下，HVT的功耗最小，LVT的功耗最大。}

5.2.2 功率预测

以SVT为阈值电压进行TSMC28nm工艺SRAM Compiler全特征的功耗仿真。

图5-17SVT下W512B8M2B1_12BIMP的全corners的DC电流值分布图

从图中可以看出图表中根据不同温度的DC数值在不同的量级，变化差异较大，故在此预测方法中对数据进行按温度分组进行回归学习与预测。

本方法将不同的温度分开进行学习。 以W80B8M2B1为目标例，

125

本次回归学习中，训练集Instances的功率数据作为特征，测试集W80B8M2B1的功率数据作为目标。 {42%：其中训练集Instances的5组corners的数据作为训练集的特征，另外10组corners的数据作为测试集的特征；} {46%：测试集W80B8M2B1的5组corners的数据作为训练集的目标，另外10组数据作为测试集的目标。} {45%：使用最小二乘法对训练集特征和目标进行回归拟合，然后将测试集的特征作为输入，} 得到测试集的目标预测值，并与测试集

的目标真实值进行对比。

本方法中的训练集的5组 Corners分别是 FG0 P72 V125 C、 FG1 P08
V125 C、 TT0 P9 V125 C、 SG0 P72 V125 C、 SG1 P08 V125
C。

图5-18W80B8M2B1的测试集目标DC的仿真值、预测值以及预测准确度

如图是本次预测的W80B8M2B1的测试集的DC的仿真值、预测值以及预测准确度。 {44%：对10组DC的预测准确度进行求平均值，得到本次学习的平均预测准确度为99.38%。} {44%：从表中也可以看出所有DC的预测准确度均处于98.5%以上，预测效果很好。}

图5-1914个目标Instances测试集的全功率预测平均准确度

上图为目标Instances测试集所有功率数据的平均准确度折线图。从折线图中可以观察到14组 Instances的测试集在10组 PVT Corners下功率数据的平均准确度都处于95.5%以上，其中所有 Instances在 AC_read和 AC_write处的预测平均准确度约为95.5%-96.5%， {43%：在 DC和 LS处的预测平均准确度约为99.6%，在 DS和 SD处的预测平均准确度分布在97%-99.5%之间。}

25

本次回归学习中，训练集Instances的功率数据作为特征，测试集W80B8M2B1的功率数据作为目标。 {42%：其中训练集Instances的5组corners的数据作为训练集的特征，另外10组corners的数据作为测试集的特征；} {46%：测试集W80B8M2B1的5组corners的数据作为训练集的目标，另外10组数据作为测试集的目标。} {45%：使用最小二乘法对训练集特征和目标进行回归拟合，然后将测试集的特征作为输入，} 得到测试集的目标预测值，并与测试集的目标真实值进行对比。

本方法中的训练集的5组 Corners分别是 FG0 P72 V25 C、 FG1 P08
V25 C、 TT0 P9 V25 C、 SG0 P72 V25 C、 SG1 P08 V25 C。

如图是本次预测的W80B8M2B1的测试集的DC的仿真值、预测值以及预测准确度。 {44%：对10组DC的预测准确度进行求平均值，得到本次学习的平均预测准确度为98.44%。} {44%：从表中也可以看出所有DC的预测准确度均处于97.5%以上，预测效果很好。}

图5-20W80B8M2B1的测试集目标DC的仿真值、预测值以及预测准确度

图5-2114个目标Instances测试集的全功率预测平均准确度

上图为目标Instances测试集所有功率数据的平均准确度折线图。从折线图中可以观察到14组 Instances的测试集在10组 PVT Corners下功率数据的平均准确度都处于96%以上，其中所有 Instances在 AC_read和 AC_write处的预测平均准确度约为99.6%，在 DC和 LS处的预测平均准确度约为98.5%~99.5%，在 DS和 SD处的预测平均准确度分布在96%~99.5%之间。

-40

本次回归学习中，训练集Instances的功率数据作为特征，测试集W80B8M2B1的功率数据作为目标。 {42%：其中训练集Instances的5组corners的数据作为训练集的特征，另外10组corners的数据作为测试集的特征；} {46%：测试集W80B8M2B1的5组corners的数据作为训

训练集的目标，另外10组数据作为测试集的目标。} {45%：使用最小二乘法对训练集特征和目标进行回归拟合，然后将测试集的特征作为输入，} 得到测试集的目标预测值，并与测试集的目标真实值进行对比。

本方法中的训练集的5组 Corners分别是 FG0 P72 VN40 C、 FG1 P08 VN40 C、 TT0 P9 VN40 C、 SG0 P72 VN40 C、 SG1 P08 VN40 C。

图5-22W80B8M2B1的测试集目标DC的仿真值、预测值以及预测准确度

如图是本次预测的W80B8M2B1的测试集的DC的仿真值、预测值以及预测准确度。 {45%：对10组DC的预测准确度进行求平均值，得到本次学习的平均预测准确度为99.81%。} {44%：从表中也可以看出所有DC的预测准确度均处于99.5%以上，预测效果很好。}

图5-2314个目标Instances测试集的全功率预测平均准确度

上图为目标Instances测试集所有功率数据的平均准确度折线图。从折线图中可以观察到14组 Instances的测试集在10组 PVT Corners下功率数据的平均准确度都处于99.2%以上，其中所有 Instances在 AC_read和 AC_write处的预测平均准确度约为99.7%，在 DC和 LS处的预测平均准确度约为99.5%~99.9%，在 DS和 SD处的预测平均准确度分布在99.2%~99.9%之间。

从以上SVT作为阈值电压的功耗数据预测中，针对Process corners的温度不同进行了分组学习。W80 B8 M2 B1中功耗 DC在125、25、-40时的最小预测准确度分别是98.5%、97.5%、99.5%；由此我们可以推测本预测方法在预测 W80 B8 M2 B1的6个功耗参数的准确度都很高，并以此为据，进行了14个目标 Instances的仿真，分别得到125、25、-40时每组的6个功耗数据的平均预测准确度，其最小平均预测准确度分别是95.5%、96%、99.2%。从预测平均准确度的结果来看，本方法应用于 TSMC28 nm工艺的 SRAM Compiler的表征功耗数据的预测具有很高的准确度， {43%：表明了通过最小二乘法进行回归学习与预测功耗数据的可行性与准确性。}

以本套SRAM Compiler为例，以50个Intel (R) Xeon 频率为3.60MHZ的4核CPU，其全corners功耗表征仿真工作量需要大致720小时。在本功耗预测方法应用于仿真工作中，本套SRAM Compiler全部仿真数据获得大概需要200小时。缩减了约72%的仿真时间。由此可见，本时序预测方法可以很有成效的减少仿真工作量，同时节省仿真时间和仿真资源。

5.3本章小结

本章首先介绍了读操作、写操作和电源模式的时序含义，提出了两种时序预测方法：相同Instances预测、不同Instances预测。其中相同Instances预测可以实现预测相同Instances不同Input slew和output loading的时序数据；不同Instance预测提供三种方法实现相同Input slew和output loading的不同Instances的时序数据预测。且以上方法都能达到97%和95.5%以上的预测平均准确度。并且分析了应用本方法后本SRAM Compiler的时序仿真能节省65%的仿真时间。

本章的第二部分介绍了功率参数的含义，提出了基于温度分类的功率参数分类方法，且本方法能达到至少95.2%、96%、99.2%的预测平均准确度。并分析了应用本方法后本SRAM Compiler的功耗仿真能节省72%的仿真时间。

第六章总结与展望

6.1 论文总结

现代 SOC 应用场景的不断拓展,对 SOC 不同的需求被提出,存储器作为 SOC 的重要部分,也对存储器提出了极高的要求,其中使用的 SRAM 在此背景下需要不断提升以适用于不同的需求,在各种高速、低功耗技术不断被提出的状态下,{45%: 其对本身就很复杂耗时的 SRAM 表征环节提出了更高的要求。}

{44%: 本文以 UMC28nmHLP 工艺的标准单元库为例,提出了两种应用于标准单元库的时序预测方法:} 电路参数预测方法和时序预测方法。使用最小二乘回归算法预测了标准单元电路的时序,其中电路参数方法的预测时序的准确度能达到 87.36%; 时序方法的预测时序的准确度能达到 98.68%。并从时间成本和机器资源成本对两种方法进行了评估,电路参数方法节约时间大致有 30%,时序方法节约时间大致有 50%。

{43%: 以标准单元库时序预测的可行性为出发点,本文还提出了针对 SRAM Compiler 表征的数据预测方法。}

本文还介绍了 SRAM 的基本框架和基本电路结构,针对影响表征仿真工作量的所有参数特征、word、bit、mux、bank、input slew、output loading 和 PVT corners 等进行了详细的阐述,针对不同仿真的需求: 相同 Instance 不同 Input slew 和 output loading 之间的预测、相同 input slew 和 output loading 下不同 Instance 之间的预测。对两种需求分别提出了一种和三种预测方法,其中针对不同 instance 之间预测的三种方法分别具有不同的应用场景。{42%: 从预测结果来看,相同 Instance 预测方法能达到最低 97% 的平均准确度;} 不同 Instance 预测方法分别能达到最低 96.99%、95.31%、96.24% 的平均准确度。从时间成本个机器成本进行了评估,本时序预测方法能节省仿真时间大约 67%。

最后对 SRAM Compiler 表征的功耗数据也提出了一种基于温度分类的功率参数分类方法,三个温度段的预测方法能达到至少 95.2%、96%、99.2% 的预测平均准确度。并分析了应用本方法后本 SRAM Compiler 的功耗仿真能节省 73% 的仿真时间。

6.2 工作展望

本文主要的工作集中在 SRAM Compiler 表征的时序与功耗的预测,本文的不足以及后续工作包括:

本课题时序数据都是选取全特征的 Instance,未对单个或多个 Instance 特征进行仿真与预测工作,接下来会对这部分进行研究。

本课题功耗数据中只针对 SVT 下的功耗数据进行了预测工作,将来需要对 HVT 和 LVT 下的功耗数据进行预测研究。

本文采用 TSMC28nmHPC+ 工艺的 SRAM Compiler 进行了表征数据预测,接下来还需要针对不同工艺的 SRAM Compiler 进行表征数据的预测。