

## 论文检测全文报告

## 基本信息

报告编号: 2022030171918570BE02744C11

文档名称: FPGA资源估算

过滤操作: 已过滤参考文献

文档作者: 张曦

已过滤自引"张曦"的相似影响

提交方式: 上传文档检测

已过滤时间, 仅与2022及之前的文献进行对比

提交时间: 2022年03月01日

正文字符数: 44181

检测范围: 大雅全文库

## 总体结论

文献相似度: 23.18%

文献原创度: 76.82%

去除参考文献相似度: 23.18%

单篇最大相似度: 3.39%

去除本人已发表论文相似度: 23.18%

单篇最大重复数: 1525

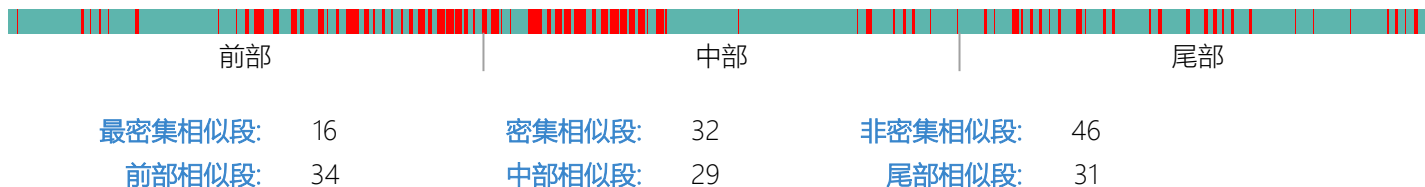
重复字符数: 10240

最相似文献题名: 精通Verilog HDL语言编程

过滤前的相似度: 23.64%

过滤前的重复字数: 10240

## 相似片段分布



## 典型相似文献

## 相似图书

序号	题名	作者	出处	相似度
1	精通Verilog HDL语言编程	刘波	北京: 电子工业出版社, 2007.05	3.39%
2	FPGA原理和结构	天野英晴;赵谦	北京: 人民邮电出版社, 2019.02	3.05%
3	机器学习实战	AurélienGéron	北京: 机械工业出版社, 2020.09	2.87%
4	机器学习实战 基于Scikit-Learn和TensorFlow	奥雷利安·杰龙	北京: 机械工业出版社, 2018.08	1.34%
5	机器学习原理及应用	陈海虹	成都: 电子科技大学出版社, 2017.07	0.79%
6	复杂情感分析方法及其应用	李勇;谢可;于卓	北京: 冶金工业出版社, 2020.02	0.72%
7	Verilog HDL硬件描述语言	杜建国	北京: 国防工业出版社, 2004.01	0.67%
8	Python机器学习算法与应用	邓立国	北京: 清华大学出版社, 2020.02	0.58%
9	现代数字图像处理	朱秀昌;唐贵进	北京: 人民邮电出版社, 2019.12	0.55%
10	EDA原理及Verilog实现	何宾	北京: 清华大学出版社, 2010.07	0.51%
11	大数据技术基础		成都: 电子科技大学出版社, 2020.09	0.47%
12	大数据处理技术	张德海;张德刚;何俊	北京: 科学出版社, 2020.04	0.47%
13	机器学习测试入门与实践	艾辉	北京: 人民邮电出版社, 2020.09	0.47%
14	EDA技术与应用	韩鹏;李岩;陈海宴	北京: 机械工业出版社, 2019.01	0.46%

15	EDA技术与应用	陈海宴;游余新;郑玉珍	北京: 机械工业出版社, 2012.05	0.46%
16	智慧城市中的大数据分析技术	秦志光;刘峤;刘瑶;钟婷	北京: 人民邮电出版社, 2015.02	0.45%
17	Hadoop大数据分析与挖掘实战	张良均	北京: 机械工业出版社, 2016.01	0.43%
18	水果品质智能分级技术	王凤云;郑纪业;刘延忠	北京: 中国农业科学技术出版社, 2018.11	0.42%
19	FPGA实战训练精粹	张晋荣;章振栋;刘荣福	北京: 清华大学出版社, 2019.02	0.42%
20	基于FPGA的SOPC嵌入式系统设计与典型实例	王刚;张激	北京: 电子工业出版社, 2009.01	0.41%
21	普通高等院校数据科学与大数据技术专业“十三五”规划教材 Python语言程序设计基础	文必龙;杨永;伏玉琛;刘志刚	武汉: 华中科技大学出版社, 2019.05	0.41%
22	CPLDFPGA与ASIC设计实践教程 第2版	陈赓	北京: 科学出版社, 2010.09	0.41%
23	FPGACPLD应用设计200例: 下册	张洪润;张亚凡	北京: 北京航空航天大学出版社, 2009.01	0.4%
24	Verilog HDL数字系统设计及其应用	袁俊泉	西安: 西安电子科技大学出版社, 2002.11	0.38%
25	机器学习从入门到精通	陈怡然	西安: 西安电子科技大学出版社, 2020.05	0.38%
26	无线通信FPGA设计	田耘;徐文波;张延伟	北京: 电子工业出版社, 2008.02	0.35%
27	CPLDFPGA与ASIC设计实践教程	陈赓;朱如琪	北京: 科学出版社, 2005.08	0.33%
28	大数据处理技术与应用	彭进香	北京: 清华大学出版社, 2020.06	0.33%
29	数字集成电路与EDA设计基础教程	陶伯睿;李静辉;苗凤娟;马丽	哈尔滨: 哈尔滨工程大学出版社, 2012.07	0.33%
30	零点起飞学Xilinx FPGA	高敬鹏;武超群;白锦良	北京: 清华大学出版社, 2019.04	0.33%
31	CMOS集成电路设计基础	孙肖子	北京: 高等教育出版社, 2008.12	0.33%
32	TensorFlow深度学习 第2版	吉安卡洛·扎克尼;礼萨·卡里姆;连晓峰	北京: 机械工业出版社, 2020.03	0.33%
33	最新集成电路设计手册 1	贾建章;武岳	银声音像出版社, 2004.03	0.33%
34	专用集成电路设计基础	孙肖子	西安: 西安电子科技大学出版社, 2003.10	0.33%
35	商务智能	吴树芳;杨国庆;朱杰	北京: 科学出版社, 2019.05	0.31%
36	Verilog HDL与数字ASIC设计基础	罗杰;肖看;谭力;柯志武;龚军	武汉: 华中科技大学出版社, 2008.03	0.28%
37	数字电子技术	林春方	北京: 高等教育出版社, 2007.06	0.28%
38	人工智能应用基础	史荧中	北京: 电子工业出版社, 2019.11	0.28%
39	R语言数据分析与挖掘	杜宾	中国铁道出版社有限公司, 2019.07	0.28%
40	数字专用集成电路的设计与验证	杨宗凯;黄建;杜旭	北京: 电子工业出版社, 2004.10	0.27%
41	电子设计案例实践基础	陈世文	西安: 西安电子科技大学出版社, 2019.10	0.26%
42	数字系统设计与Verilog HDL 第4版	王金明	北京: 电子工业出版社, 2011.01	0.26%
43	数字系统设计与Verilog HDL Vivado版	王金明	北京: 电子工业出版社, 2020.03	0.26%
44	数字电路设计 仿真 测试 第2版	余新平;覃洪英;吴爱平;郑恭明;王晓爽;付青青;刘开健	武汉: 华中科技大学出版社, 2017.12	0.25%
45	人工智能导论	刘刚;张果峰;周庆国	北京: 北京邮电大学出版社, 2020.07	0.25%
46	智能会计信息系统构建研究	续慧泓	北京: 中国财政经济出版社, 2019.12	0.25%
47	网络安全传输与管控技术	兰巨龙;江逸茗;胡宇翔;刘文芬;李玉峰;张建辉;邬江兴	北京: 人民邮电出版社, 2016.09	0.24%
48	基于Verilog HDL的数字系统设计快速入门	高敬鹏;武超群	北京: 电子工业出版社, 2016.08	0.24%
49	试验设计与数据处理	牛长山;徐通模	西安: 西安交通大学出版社, 1988.06	0.24%



50	试验设计与数据处理	郑少华;姜奉华	北京: 中国建材工业出版社, 2004.03	0.24%
----	-----------	---------	------------------------	-------

## 相似期刊

序号	题名	作者	出处	相似度
1	基于神经网络的电源出力同时率研究	廖云华	装备维修技术, 2020, 第24期	0.55%
2	Verilog HDL在电梯控制中的应用分析	陈炳贤	西南民族大学学报(自然科学版), 2014, 第A1期	0.5%
3	大数据的思维革命及算法挑战	陈一听	信息通信技术, 2013, 第6期	0.49%
4	基于随机森林的短时临近降雨预报方法*	钟海燕;李玲;麦雄发;王雨	广西师范学院学报(自然科学版), 2018, 第4期	0.47%
5	基于机器学习的织物疵点检测	王帅;岳鹏飞;董晗睿;侯爽;余灵婕	纺织科技进展, 2020, 第10期	0.47%
6	基于非视距鉴别的超宽带室内定位算法	曾玲;彭程;刘恒	计算机应用, 2018, 第A1期	0.38%
7	电信运营商物联网客户运营分析系统的构建及应用	张勋;翟春辉;张呈宇;贾捷;龙岳	电信科学, 2019, 第10期	0.36%
8	机器学习在物联网虚假用户识别中的运用	张溶芳;许丹丹;王元光;潘思宇;李正茂	电信科学, 2019, 第7期	0.36%
9	基于机器学习方法的个人信用风险评估研究①	王铭	科技创新导报, 2020, 第30期	0.36%
10	一种基于改进AHP的电信企业信用评分模型*	李金柱;唐霞;余晨;彭依校	通信技术, 2017, 第11期	0.36%
11	基于随机森林的联合收割机故障诊断技术*	刘景;杨光友;马志艳;陈学海	中国农机化学报, 2019, 第10期	0.36%
12	基于随机森林的高分辨率PM <sub>10</sub> 和PM <sub>2.5</sub> 遥感反演——以广东省为例	申原;陈朝亮;钱静;刘军	集成技术, 2018, 第3期	0.36%
13	改进的GDBT迭代决策树分类算法及其应用	曹颖超	科技视界, 2017, 第12期	0.35%
14	基于数据挖掘的互联网金融智能反欺诈业务研究	孟洁;张洪涛	经济与社会发展研究, 2021, 第18期	0.34%
15	基于Filter模型和随机森林的电信客户投诉分类方法	岳丹阳;方帅	计算机应用与软件, 2020, 第7期	0.3%
16	气流后向轨迹和门限重复单元的PM <sub>2.5</sub> 预报	梁世文;李琦;侯俊雄;冯道	测绘科学, 2020, 第3期	0.26%
17	基于周期性ARMA-SVR模型的空调冷热负荷预测	甘中学;喻想想;许裕栗;李德伟	控制工程, 2020, 第2期	0.26%
18	机器学习在心理测量中的应用	王一溢;占继尔;陈泽龙;田渊明	电脑知识与技术, 2021, 第3期	0.25%
19	基于MODIS数据的沈阳市PM <sub>10</sub> 和PM <sub>2.5</sub> 反演研究	陈美娟;刘静;刘利民;孙欣;朱明宇;蒋姗姗;乔延艳	环境科学与技术, 2016, 第12期	0.25%
20	机器学习在建筑节能中的应用	谢雨桐	城市住宅, 2020, 第12期	0.25%
21	基于半监督学习方法的磨煤机故障预警	肖黎;罗嘉;欧阳春明	热力发电, 2019, 第4期	0.24%
22	基于混合刺激诱发放松状态的方法研究	伍能彪;朱珍民;王毅	计算机与数字工程, 2019, 第10期	0.23%
23	工业机器人主动安全控制技术研究现状与展望	杨琨;夏信凯;马环洲;桑胜波	机器人技术与应用, 2021, 第3期	0.22%
24	基于Cyclone II EP2C20的8051 IP核的应用研究	樊峰;陈西曲;叶艳艳	武汉轻工大学学报, 2012, 第4期	0.22%
25	再分解优化设计方法	罗晓春;林争辉	计算机工程, 2001, 第5期	0.21%
26	基于混合变异GOA优化极大似然的DOA估计研究	陈婷;白艳萍	河南科学, 2021, 第1期	0.19%
27	融合人工鱼群和随机森林算法的膝关节接触力预测	卢巍;朱业安;徐唯祎	中国医学物理学杂志, 2020, 第4期	0.19%

28	运用机器学习的方法进行城市防洪模型研究	夏勇;贡斌;蒋占军;徐小军	电子技术与软件工程, 2020, 第4期	0.19%
29	新一代人工智能在矿山充填中的应用综述与展望	齐冲冲;杨星雨;李桂臣;陈秋松;孙元田	煤炭学报, 2021, 第2期	0.18%
30	一个基于随机森林的微博转发预测算法	罗知林;陈挺;蔡皖东	计算机科学, 2014, 第4期	0.18%
31	基于心率和呼吸特征结合的睡眠分期研究	冯静达;焦学军;李启杰;郭娅美;杨涵钧;楚洪祚	航天医学与医学工程, 2020, 第2期	0.18%
32	应战条件下基于大数据的物资协同保障模式研究	杨灵运;王春晨;张宇哲;袁江远	装备维修技术, 2021, 第1期	0.18%
33	基于随机森林的僵尸网络流量检测	肖琦;苏开宇	微电子学与计算机, 2019, 第3期	0.17%
34	基于城市道路卡口数据的交通流量预测	李浩;张杉;曹斌;范菁	重庆大学学报, 2020, 第11期	0.17%
35	基于机器学习的雷达电子侦察中的应用探析	徐福富;刘欣怡;高希光	信息记录材料, 2021, 第4期	0.16%
36	风电功率预测技术研究综述	黎静华;桑川川;甘一夫;潘毅	现代电力, 2017, 第3期	0.16%
37	一种基于气象数据的仓储粮堆表层温度预测方法	段珊珊;杨卫东;肖乐;张元	中国粮油学报, 2020, 第2期	0.15%
38	多级逻辑面积优化设计方法	罗晓春;林争辉	计算机工程与应用, 2002, 第3期	0.15%
39	红外成像导引头抗干扰性能评估方法研究	马潮;陆志洋;余海鸣;洪泽华;杨杰;乔宇	空天防御, 2018, 第4期	0.15%
40	基于心电与呼吸信号的睡眠分期算法研究	黄文汉;张伟;胡立刚;周陈旺;向丹阳	智能计算机与应用, 2018, 第1期	0.15%
41	基于ARIMA模型与BP神经网络的三七价格预测	刘建中;李慧超	计算机科学与应用, 2017, 第7期	0.15%
42	基于随机风场概率加权的台风灾害下输电线路损毁预警	侯慧;于士文;李显强;王红斌;黄勇;吴细秀	电力系统自动化, 2021, 第7期	0.15%
43	一款多核处理器FPGA验证平台的设计与实现	朱英;陈诚;许晓红;李彦哲	计算机研究与发展, 2014, 第6期	0.14%
44	基于学习大数据的学习评价: 原理、模型及实施策略研究*	蔡宝来;孙昊琛	中国教育信息化, 2021, 第3期	0.14%
45	逻辑再优化设计方法	罗晓春;林争辉	计算机自动测量与控制, 2001, 第3期	0.14%
46	基于集成学习的DoS攻击流量检测技术	马泽文;刘洋;徐洪平;易航	信息网络安全, 2019, 第9期	0.13%
47	基于大数据和机器算法的规划需求价值评估研究	胡博;赵兰奇;张丹	电信工程技术与标准化, 2018, 第4期	0.13%
48	速度优化再分解设计方法	罗晓春;林争辉	微电子学, 2001, 第4期	0.13%
49	数字分析在学生行为中的应用研究	何小儒;李玲;俞献林	教育科学论坛, 2016, 第10期	0.13%
50	一种改进随机森林算法及其在入侵检测中的应用	江轲;张宏进	电子设计工程, 2021, 第22期	0.13%

## 相似报纸

序号	题名	作者	出处	相似度
1	科学家探索AI“选择性失忆”新问题		北方新报, 2021.10.20	0.09%

## 相似网络文档

序号	题名	作者	相似度
1	逻辑综合中的工艺映射算法研究	罗晓春	1.16%
2	逻辑综合中工艺映射的研究	张镭	0.89%
3	基于doc2vec和SVM的舆情情感分析系统的研究与设计	甘如飴	0.64%
4	锁存器与触发器		0.48%





5	触发器与锁存器		0.48%
6	触发器与锁存器		0.48%
7	基于LUT结构的FPGA的工艺映射算法的研究	杨劲秋	0.48%
8	基于森林资源连续清查数据的浙江省杉木林立地质量评价	杨海宾	0.45%
9	基于字符级卷积神经网络的中文文本分类研究	刘坤	0.44%
10	开关电源控制芯片可测性电路的设计	赵倡申	0.42%
11	基于Kinect的手势图像分类器设计	高扬	0.41%
12	基于电子产品领域商品评论的情感分析方法研究和实现	文金	0.41%
13	MEMS IMUGPS组合导航滤波算法研究	刘彦博	0.4%
14	基于数据挖掘技术的交通状态判别算法研究与应用	刘英	0.37%
15	基于集成学习的高维稀疏多标签文本分类	程龙	0.36%
16	基于BP神经网络的光电容脉搏波血压监测技术研究	雷苏力	0.34%
17	近红外光谱分类的深度森林方法及应用研究	杜师帅	0.34%
18	光泵CH <sub>2</sub> subgt;2lt;subgt;Flt;subgt;2lt;subgt;气体太赫兹激光器及稳定控制的研究	戴光	0.33%
19	顶吹熔炼系统寿命预测及维修策略研究	薛宇涛	0.33%
20	联合主被动遥感数据估算森林生物量空间分布及其与陆表温度的关系	沈文娟	0.33%
21	工程物探方法探测覆盖层的工程地质效果分析	李颖	0.33%
22	面向移动基带SoC芯片前端功能、性能和效能验证的监测系统设计	王卫凯	0.32%
23	基于FPGA的100Gbps光传输数字信号处理器验证平台设计	罗文涛	0.31%
24	高效高速CMOS锁相环关键技术研究	周宇	0.31%
25	基于Android的智能车联网终端设计	姜梦婷	0.31%
26	大数据建模在保险电销系统中的应用	盖小睿	0.3%
27	基于FinFET结构小型基本单元库的开发与评估	朱刚	0.29%
28	红外焦平面探测器信号预处理电路设计	陶海洋	0.29%
29	固体继电器热分析及研究	曾庆国	0.29%
30	基于FPGA和LABVIEW的太阳能自动跟踪系统研究与仿真	张宏伟	0.29%
31	基于UVM的WIFI数字基带验证方法的研究	周轲	0.28%

## 全文对比

### 摘要

随着后摩尔时代的到来，芯片制造工艺的提升以及市场对芯片功能需求的不断提升，导致现阶段芯片设计规模越来越大。在超大规模集成电路设计阶段验证已经逐渐成为困扰各大芯片设计人员的关键问题，虽然可以使用软件仿真、硬件加速仿真等验证方法来加速验证流程，但是随着集成电路设计规模逐渐增大，原先的验证方法在时间成本上已经无法满足当前快速设计迭代的需求。随着FPGA（可编程逻辑阵列）的快速发展，藉由FPGA的可擦除、可组网、可快速模仿集成电路设计行为等特性，使用FPGA进行芯片设计原型验证已逐渐成为验证阶段主流。但随着设计的规模剧增，单片FPGA已无法满足超大型集成电路设计的验证需求，从而衍生出高密度FPGA原型验证系统。而如何将超大型集成电路设计合理的划分至高密度FPGA原型验证系统中的每个FPGA中，将成为集成电路验证划分阶段的关键问题，而超大型集成电路设计所需要的FPGA片上资源数量将作为合理划分电路的重要依据。同时随着机器学习的快速发展，将EDA软件与机器学习进行结合从而提升时间性能也逐步成为EDA行业的发展趋势。

本文选题主要针对高密度FPGA原型验证系统中划分阶段对集成电路如何进行快速资源估算并基于相关验证阶段需求与原理结合机器学习展开具体的研究与方法尝试。论文的主要工作与成果如下：

(1) 对原型验证流程中综合阶段进行基本研究，并举例简述在逻辑综合的基本原则及技术映射阶段的常见算法，以此为依据使用开源软件Verific作为基础开发传统意义上的快速资源估算模型，并使用来自多个行业的超大规模集成电路进行验证，在未进行优化阶段时间提升超过

100%，对LUT与FF资源平均估算准确度达60%以上，验证并实现了使用传统方法进行快速资源估算模型。

(2) 上述传统快速资源估算模型的基础上加入合理的优化算法，增加为满足于高密度FPGA原型验证系统划分阶段需求的BottomToUp算法，使资源估算时间优化较优化前时间利用率提升达220%以上，实现并增加LUT合并、面积约束等优化及约束项，从而提升快速资源估算模型精度在不使用特殊器件情况下对LUT及FF资源估算最高准确率可达90.11%，从而进一步提升了快速资源估算模型的性能及可实用性。

(3) 研究人工智能与传统EDA工具相结合的可能性，并使用Verific为基础，通过进一步分析逻辑综合与技术映射的原理，分析集成电路RTL级描述语言特征，构建RTL级描述特征提取器，实现提取RTL级描述特征67种。对特征相关性、贡献度进行进一步分析后进行相关特征工程，结合随机森林、SVM支持向量机、神经网络构造对应的RTL级设计所需FPGA片上资源估算学习模型，并完成网络参数调校及训练，实现平均LUT、FF、DSP等资源估算模型确定系数为73.4%的神经网络模型。最后通过对其数据进行分析，进一步验证了机器学习在原型验证划分阶段集成电路快速资源估算领域的可行性。

关键词：FPGA原型验证，资源估算，特征工程，机器学习

硕士学位

ABSTRACT

插图索引

图1.1 高密度FPGA原型验证系统 2

图1.2 RTL Partition流程 3

图1.3 2008年Xilinx公司快速资源估算模型 5

图2.1 FPGA基本组成 8

图2.2 LUT结构图示 9

图2.3 LUT基本工作原理 9

图2.4 D触发器 10

图2.5 FPGA设计基本流程 11

图2.6 FPGA逻辑综合流程 13

图2.7 连续赋值语句综合图示 14

图2.8 阻塞赋值语句综合图例 14

图2.9 非阻塞赋值语句综合图例 15

图2.10 非阻塞赋值语句综合图例 15

图2.11 算数运算符综合图例 15

图2.12 if语句综合图例 16

图2.13 case语句综合图例 16

图2.14 循环语句综合图例 17

图2.15 函数语句综合图例 17

图2.16 工艺映射流程图 18



图2.17 Mapping算法伪代码 20

图2.18 电设计转化为DAG图 20

图2.19 逻辑划分 21

图2.20 器件覆盖 21

图2.21 机器学习实现RTL级设计资源估算流程 22

图2.22 Bagging算法伪代码 24

图2.23 决策树学习基本算法 25

图2.24 随机森林模型算法步骤 27

图2.25 神经元计算图示 28

图2.26 感知器结构图示 29

图2.27 多层感知机结构图示 30

图3.1 VeriTreeNode类视图 36

图3.2 VeriModule类视图 37

图3.3 Verific-Netlist-DataBase数据结构图 38

图3.4 快速资源估算模型文件流 39

图3.5 快速资源估算模型流程图示 40

图3.6 快速资源估算模型Parse-Design流程 41

图3.7 RTL Elaborate操作图示 41

图3.8 Flatten操作电路图示 42

图3.9 Flatten操作树结构图示 42

图3.10 RTL级基本器件库 43

图3.11 快速资源估算模型计算结果 43

图3.12 RTL代码中的Glue Logic和Instance 44

图3.13 BottomToUp算法数据结构 45

图3.14 BottomToUp算法伪代码 46

图3.15 优化后快速资源估算模型输出 47

图3.16 LUT-5 Base FPGA结构图示 48

图3.17 4输入查找表真值表 49

图4.1 特征提取器结构图 54

图4.2 Verilog操作符图示 55



图4.3 运算符特征提取伪代码	56
图4.4 连续赋值语句特征提取伪代码	57
图4.5 Verilog Always过程块语法格式	57
图4.6 Always特征提取伪代码	58
图4.7 Verilog过程赋值语句语法格式	59
图4.8 Verilog If条件语句语法格式	60
图4.9 If条件语句特征提取伪代码	60
图4.10 Verilog Case语句语法格式	61
图4.11 Case条件分支语句特征提取伪代码	61
图4.12 Verilog循环语句语法格式	62
图4.13 循环语句特征提取伪代码	63
图4.14 Verilog HDL Function语法结构	63
图4.15 RTL语法特征提取器流程图	64
图5.1 特征数据集直方图	68
图5.2 特征热力图	69
图5.3 特征预处理流水线	73
图5.4 模型训练开发流程	74
图5.5 各模型尝试预测LUT资源结果	75
图5.6 机器学习模型训练及调优流程	76
图5.7 随机森林预测LUT模型参数搜索学习曲线	76
图5.8 随机森林LUT预测模型学习曲线	77
图5.9 随机森林预测FF特征搜索学习曲线	77
图5.10 随机森林FF预测模型学习曲线	78
图5.11 神经网络学习曲线，左图为超参数调优前学习曲线	78
图5.12 神经网络模型结构图	79
图5.13 神经网络与LUT模型学习曲线	79
图5.14 神经网络模型结构图	80
图5.15 神经网络模型预测FF资源学习曲线	80
图5.16 机器学习模型验证结果	81





表2.1 典型多层感知机架构 31

表3.1 快速资源估算模型无优化结果对比 48

表3.2 快速资源估算模型增加优化结果对比 48

表4.1 Verilog主要语法类型与关键字 51

表5.1 部分基本数据集数据图示 65

表5.2 数据集基本概要 66

表5.3 RTL级特征与LUT资源的相关性数据 68

表5.4 RTL级特征与FF资源的相关性数据 68

表5.5 RTL级特征与BUFG资源的相关性数据 69

表5.6 RTL级特征与IO资源的相关性数据 69

表5.7 模型评估 79

符号对照表

符号 符号名称

激活函数

损失函数

特征权值分数

熵

特征个数

梯度

二维高斯函数

保留的特征个数

M 正交基向量

Q 协方差矩阵

Z 加权和

缩略语对照表

缩略语 英文全称 中文对照

ASIC Application Specific Integrated Circuit 专用集成电路

SoC System on Chip 片上系统

FPGA Field Programmable Gate Array 现场可编程逻辑门阵列

RTL Register Transfer Level 寄存器传输级别



LUT Look Up Table 显示查找表

DSP Digital Signal Processing 数字信号处理单元

BRAM Block RAM 块状存储

HLS High Level Sythesis 高层次综合

LB Logic Block 逻辑块

FF Flip-Flop 触发器

CLB Configurable Logic Block 可配置逻辑块

RAM Random Access Memory 随机存取存储器

SRAM Static Random Access Memory 静态随机存取存储器

HDL Hardware Description Language 硬件描述语言

API Application Programming Interface 应用程序接口

CAD Computer Aided Design 计算机辅助设计

EDA Electronic Design Automation 电子设计自动化

RF Random Forest 随机森林

LR Linear Regression 线性回归

DT Decision Tree 决策树

SVM Support Vector Machine 支持向量机

MLP Multilayer Perceptron 多层感知机

ANN Artificial Neural Network 人工神经网络

硕士学位

目录

摘要.....I

ABSTRACT.....III

插图索引.....V

表格索引.....IX

符号对照表.....XI

缩略语对照表.....XII

目录.....XIII

第一章绪论.....1

1.1.....研究背景.....1



1.2.....	国内外研究现状.....	4
1.3.....	研究的意义.....	5
1.4.....	主要内容及组织结构.....	6
第二章.....	FPGA资源与估计方法.....	7
2.1.....	FPGA基本结构概述.....	7
2.1.1.....	FPGA的资源组成.....	7
2.2.....	FPGA综合原理概述.....	10
2.2.1.....	VerilogVHDL.....	10
2.2.2.....	基于FPGA综合与映射的资源计算方法.....	12
2.2.3.....	FPGA技术映射.....	17
2.3.....	机器学习资源估计方法与模型.....	21
2.3.1.....	机器学习与EDA结合的可行性.....	22
2.3.2.....	特征提取方法.....	22
2.3.3.....	随机森林.....	23
2.3.4.....	人工神经网络.....	28
2.4.....	机器学习性能评估指标.....	31
第三章.....	快速资源估算模型.....	35
3.1.....	Verific及相关数据结构.....	35
3.1.1.....	Verific平台介绍.....	35
3.1.2.....	Parse-Tree数据结构.....	35
3.1.3.....	Netlist数据结构.....	37
3.2.....	模型构建.....	39
3.2.1.....	模型流程分析.....	39
3.2.2.....	建模实现.....	40
3.3.....	模型优化.....	44
3.3.1.....	BottomToUP算法.....	44
3.3.2.....	冗余逻辑优化及合并.....	47
3.3.3.....	最小面积优化方法.....	48
3.4.....	结果分析.....	50
3.5.....	本章小结.....	51



第四章.....特征提取.....	53
4.1.....VerilogHDL语法特征分析.....	53
4.2.....特征提取方法.....	54
4.2.1.....端口信息.....	54
4.2.2.....线网信息.....	55
4.2.3.....操作符及运算表达式.....	55
4.2.4.....连续赋值语句.....	56
4.2.5.....过程结构体.....	57
4.2.6.....过程赋值语句.....	59
4.2.7.....If-else条件分支语句.....	59
4.2.8.....Case条件分支语句.....	60
4.2.9.....循环语句.....	61
4.2.10.....函数语句.....	63
4.3.....特征器构建.....	64
4.4.....本章小结.....	65
第五章.....构建机器学习模型及训练.....	67
5.1.....数据集分析.....	67
5.1.1.....数据集构建.....	67
5.1.2.....数据集分析.....	68
5.2.....特征工程.....	72
5.2.1.....特征预处理.....	72
5.2.2.....转换流水线.....	73
5.3.....模型框架选择.....	74
5.3.1.....初步模型选择.....	74
5.4.....模型训练.....	75
5.4.2.....随机森林.....	76
5.4.3.....人工神经网络.....	78
5.5.....结果与分析.....	81
5.6.....本章小结.....	82
第六章.....总结与展望.....	83



6.1.....	工作总结.....	83
6.2.....	工作展望.....	84
参考文献.....		85
致谢.....		89
作者简介.....		91

## 绪论

### 研究背景

随着市场环境的逐步升级，市场需求对与IC的功能愈加严格，IC的复杂度在工艺的进步下也逐步增加。目前的ASIC（Application Specific Integrated Circuit，专用集成电路）设计，这项又称为片上系统SoC（System on Chip，片上系统）的集成电路技术在业界已经得到广泛的认可与应用，且当代的SoC设计也不在局限于一块IC上一般情况下，由于采用ASIC设计具有功耗低、投产成本低、迭代周期快等优势，所以ASIC可容纳更为复杂和灵活的逻辑设计。同样正是由于现代ASIC设计的高复杂度，使得其研发设计的风险也在大幅增加。一旦芯片设计失败，将会导致大量研发本的损失，所以，IC正确性验证逐步变为超高复杂度IC研发过程中必备且关键的环节。最新一项统计数据表明在一些项目中芯片的设计团队规模甚至不如验证团队规模，验证所花费的时间也占到产品研发周期的一半以上，找到有效可行的验证方法成为验证的关键[4]。目前，最常见的验证方法是软件仿真技术，但是软件仿真虽然简单易行，在面对复杂的集成电路设计时，其验证效率过低，最终可能影响到产品的及时发布[6]。而基于FPGA的原型验证方法，依靠其可反复擦除、可真实的模拟设计中的硬件行为以及能够系统全面的反映设计功能等优势，被越来越多的验证团队所采用[6]。

目前随着高性能ASIC芯片设计复杂度的提升以及体系结构的不断升级换代及创新，SoC设计的规模在快速膨胀，正确性验证已成为高性能芯片研发过程中的关键环节，因此为保证验证的高效性和准确性许多EDA或集成电路设计公司相继推出了许多相对高级的验证方法。在具有超高复杂与高性能IC验证上，主流验证方法分为以下三类：软件模拟验证、硬件加速器仿真验证、FPGA原型验证。软件模拟验证是验证集成电路设计功能正确性上最常采用的方法，其具有模拟精确、方便灵活等优点，因此对于量级较小的IC模块的正确性验证通常会采用EMU的方法。但是，当芯片整体复杂度提升，软件模拟的效率也在不断下降，同时也将难以覆盖整体IC功能的验证。实践表明，不论使用哪种算法去设计EDA工具的仿真方式，当IC设计所包含的电路门数超过一定数目时，EDA工具对于系统信号的反应将很难保证，验证的效率也将会被极大制约，从而很难高效的实现验证结果[3]。针对软件模拟仿真的局限性，各大EDA厂商与集成电路厂商进而提出通过硬件加速器来实现仿真验证，其主要思想是通过专门的硬件加速器，将软件仿真中的一些复杂运算通过使用告诉硬件加速器来计算，最后硬件加速器将运算结果传回仿真软件，最终提升整体上的验证效率。而且硬件加速器验证的仿真环境也更加真实，相比于完全使用软件来实现仿真，硬件加速器使用了真实的硬件仿真环境，可以更加真实的反映集成电路设计系统的各种问题，从而可以辅助开发人员有效的发现软件仿真无法捕获的异常，这对于验证工程师的调试很有帮助，典型的硬件仿真器有Cadence公司的Palladium和Mentor公司的Veloce系列[35]。但是硬件仿真器也存在固有的缺陷，因为其需要针对特定的电路构造特点的硬件加速设备，同时还需要EDA软件与硬件加速设备间的通信接口，而一旦所验证的设计改变，原有的硬件加速固件将可能无法复用，从而需要重新设计硬件固件及相关接口。从理论上讲，本文的所提及的高密度原型验证系统亦可基于硬件仿真器实现，但硬件仿真器的成本高昂，而且由于当代先进FPGA的技术发展及配套软件已经较为完备，所以通过基于FPGA可反复擦除的特性，使用先进FPGA来进行高密度原型验证成为了当下集成电路设计验证阶段较优的选择。随着FPGA性能的不不断提升，基于FPGA的原型验证逐渐成为大规模复杂芯片验证的选择。

### 高密度FPGA原型验证系统

近年来，随着人工智能、GPU、CPU等高性能模块的快速发展与运算需求的增加，仅靠单个FPGA的片上资源已难以ASIC芯片设计的需要，因此基于多颗FPGA的高密度原型验证技术将成为最优的选择。

对于能够使用多FPGA进行正确的验证，则必然需要进行逻辑划分，而为实现正确且合理的逻辑划分，首先要解决的就是设计逻辑综合时间长的的问题[4]。一个设计规模达到几亿门甚至数十亿门的大规模SoC设计，若使用整个设计先进行综合，再进行逻辑分割的方式，设计的综合时间会非常的漫长，一次综合流程就有可能花费的时间长达数天甚至数周，若设计的规模特别大，甚至会出现编译软件或服务器崩溃的情况[5]。显然，面对超大规模的SoC设计采用RTL（Register Transfer Level，寄存器传输级别）级分割，将会成为最优的选择，RTL-Partition流程如图1.2所示。

### 高密度FPGA原型验证系统流程

使用RTL划分通过在RTL阶段将设计划分为个子设计后，便可以有效解决设计逻辑综合时间长的的问题，同时实现并行编译、综合、布局布线的

流程，从而有效提升FPGA原型验证的效率。但为正确实现RTL级的逻辑划分，则需要解决以下几个关键问题：

(1) 首先，在对超大型SoC设计不进行逻辑综合前提下进行RTL级别分割时，需确定最合理的逻辑划分的边界，满足SoC设计的全局时钟树，同时使得进行逻辑划分后的SoC设计满足整个大型FPGA原型验证系统的互联资源和每个划分后的子设计不超出每个FPGA可综合的资源数量（为增加FPGA综合及布局布线的成功率，常使用FPGA资源总量的60%-70%）；

(2) 其次，由于无法对设计进行整体综合，则无法获得设计在FPGA上所需的准确片上资源数量，则如何对设计在RTL级进行快速资源估算，在不进行逻辑综合为前提获得设计中每个模块的资源，从而为确认划分的边界提供参考依据将成为关键问题；

(3) 最后，如何通过已经获得的RTL级设计的资源与大型原型FPGA设计平台相结合，能同时满足FPGA间的互联结构、互联资源、片上逻辑资源等条件实现合理的逻辑划分，并且划分结果与原始设计逻辑等价；

本文将主要讨论上述问题中的（1）（2），即如何对RTL设计进行快速且满足RTL级逻辑划分所需精度的快速逻辑资源估算方法并尝试结合机器学习探究解决该方法的方法。

#### 国内外研究现状

当设计者通过硬件描述语言完成对集成电路寄存器传输级（Register Transfer Level, RTL）的软件编码设计后，RTL级设计将通过几个基本步骤，最终转化为用于配置FPGA内部结构的比特流文件。RTL需要经过的基本设计步骤，包括逻辑综合、逻辑优化、技术映射、逻辑打包、布局、布线等。

FPGA的硬件组成结构中包含许多不同类别的片上资源，主要有实现逻辑电路所需要的逻辑资源，用于提供外部输入输出接口的资源，以及连接并导通前两种资源的布线资源。具体来讲，一般包括触发器（F-F）、数字处理单元（DSP）、查找表（LUT）、块状存储（Block RAM, BRAM）等多种片上起源。其中，LUT和FF等逻辑资源一般存在于逻辑块LB（Logic Block或slice）中。

由于高密度FPGA原型验证整个流程如图1.2包含多个步骤，即RTL级设计组网布局、RTL级设计分割、逻辑综合、工艺映射、单个FPGA的布局和布线等，由于高密度FPGA原型验证流程步骤繁多且耗时较长，设计者在编写和修改RTL描述时，无法快速预估待划分电路在FPGA上实现时需要的片上硬件资源数量，这样就可能导致无法完成综合流程或者无法完成FPGA布局布线，或互联关系复杂，无法满足FPGA间组网关系，从而需要多次设计迭代来解决上述问题。另外，在逻辑综合流程中，每一部分RTL级描述所需要的实际FPGA片上资源的数量，也将对RTL到门级网表的转化与优化过程有指导意义。

综上所述，在RTL级分割阶段如何能够快速且准确的得到RTL级描述所需要的FPGA片上资源使用量，是一个很关键且重要的问题。虽然在国内也有部分针对大型设计提供EDA及FPGA原型验证厂家，但是对于在划分阶段涉及的资源估算问题在我国国内暂时没有可商业使用的较成熟的解决方案。

随着FPGA设计规模的越来越大，对于片上资源估算这一环节的重要性越发明显，也是必不可少的步骤，使得FPGA资源估算这一问题也逐渐成为学者和EDA厂商的研究重点，随之而来的也是多种多样的估算方法也都各有优劣。

2008年Xilinx公司Paul Schumacher 和 Pradip Jha提出由于用户在对FPGA设计的开发使用过程中，无法规避对设计进行逻辑综合这一步骤，从而可以通过使用快速逻辑综合生成的网表结构，通过在设计的宏观层面对现有的器件匹配库对原设计进行匹配替换，从而得到初步的资源估算结果，使用该方法可以较快速的得到不计精度的资源粗略估计值[17]。

#### 2008年Xilinx公司快速资源估算模型[17]

2011年代尔夫特工业大学Roel Meeuws等人提出可以通过深度学习建立基于统计方法的高级定量硬件预测模型设想，并做出了初步的验证，为人工智能在资源估计这一问题上的实现建立可行性分析[11]。

2017年法国瓦伦西亚大学Mariem MakniLAMIH发布的论文中提出一种面向异构FPGA的SoC片上资源估算方法，其出发点是通过结合HSL（High Level Synthesis，高层次综合）报告构建模型对布局后FPGA真实面积进行估算[13]。

2018年Steve Dai, Yuan Zhou等人在IEEE会议上发表的论文中提出了一种通过提取HLS报告相关信息特征对FPGA布局布线后资源及时序进行预测的方法，并且达到了一个较好的预测结果，由于HLS是比RTL级在设计流程上更高层级、更抽象的表述方法，所以该方法的提出进一步为通过RTL级特征预测实际所需FPGA片上资源提供了可行性[12]。

通过对国内外研究现状的分析，我们可以发现在面向高密度FPGA原型验证系统的RTL级资源估算方面我国还存在较大的待填补的空白。





## 研究的意义

对于高密度FPGA原型验证系统而言，面向的设计都是规模上千万门或设计规模超过亿门，更有甚者规模超过百亿门，需要上百片当代先进FPGA才可容纳下整个设计，但是对于给定的RTL描述，它所需的FPGA片上硬件资源数量一般需要在完成逻辑打包和布线后，才可以全部得到。但对于规模较大的电路设计，从RTL描述到布线的设计综合流程往往需要花费几个小时乃至数十小时。当出现片上资源不足的问题时，设计者需要修改RTL描述并进行耗时较长的设计综合流程迭代。

若用户的设计超过百亿门时，进行一次迭代所耗费的时间成本，人力成本都是不可预估的，所以若可以在不进行综合、布局布线等流程时快速获得所需要的所需要资源的估计值，将会为缩短开发人员的迭代周期提供帮助。同时由于高密度FPGA原型验证划分阶段需要以设计资源使用情况来划分合理的分割边界，合理的分割边界将会对设计在满足硬件FPGA组网前提下为系统的运行，规划，验证，调试提供良好的帮助，而分割边界的确认需要以RTL设计中的每个层级中每个节点的Instance的各项资源作为依据。

综上所述不论对于开发人员还是对于高密度FPGA原型系统而言，能够快速且精确的获取到大型设计所需的FPGA片上资源及对应层级所需的资源的估计值将具有巨大的意义。

## 主要内容及组织结构

针对我国需要开发具有自主知识产权的大型FPGA原型验证流程和相应的支持EDA软件，本文首先介绍了FPGA的基本结构及大型FPGA原型验证流程中的逻辑综合、技术映射与机器学习模型的一般方法作了详细描述，最后分别以上述基本流程为基础实现了以传统流程为基础的快速资源估算模型及以机器学习为基础的RTL级资源估算模型，**两个模型的主要工作体现在以下几个方面：**

创建快速资源估算模型的基本方法与方案概述。

基于C++设计并实现面向工程的基于传统流程的快速资源估算模型。

在快速资源估算模型的基础上引入更符合高密度FPGA原型验证系统的算法、优化及约束项，进一步提升其精度与性能。

设计RTL级面向资源估算的语法特征提取及基于C++的代码实现。

基于机器学习的RTL级资源估算的方法研究及分析。

## FPGA资源与估计方法

**FPGA是可编程逻辑器件（PLD）的一种，是可以用来实现任意逻辑电路的集成电路。**FPGA的出现是当代集成电路快速发展的一个缩影。上世纪，由于研发、设计集成电路的成本急剧增加，集成电路研发企业都想要找到一种即满足功能需求又价格低廉的芯片。在这种时代背景下，现场可编程门阵列FPGA应运而生。相对于其他相同种类的芯片，由于其可反复擦除的特性，使得基于FPGA进行研发的容错率更好，成本降低。在目前的多数集成电路研发团队的研究过程中，研发团队会先基于一定数量的FPGA芯片对集成电路功能、算法进行实现，并根据实际情况进行修改直到设计达到预期的效果，最后再将达到要求的设计用ASIC实现并最终投入生产。

**FPGA的特征从名字就可以看出是在现场可编程的门阵列。但实际上FPGA并非是单纯由“门”形成的结构。**本章将主要介绍FPGA的基本构成资源及大型FPGA原型验证的基本流程。

## FPGA基本结构与资源组成

FPGA大致由三部分构成：第一部分是实现逻辑电路的逻辑要素即逻辑块（Logic Block，LB）也就是我们常说的基本逻辑单元，可编程逻辑中逻辑块的实现方式有多重，比如最早的乘积项和现在的查找表、数据选择器等，但是无论哪种方式都是由可以实现任意逻辑电路的可编程部分触发器即Flip-Flop，FF等数据存储电路和数据选择器组成。逻辑要素是在FPGA设计开发流程中最为紧俏的资源种类之一，FPGA供应商对各自的逻辑块结构有不同的称呼，如Xilinx公司称其为CLB(Configurable Logic Block)，采用的是查找表的方式，但基本原理都是类似的。第二部分是和外部进行信号输入、输出对的要素即（IO块），第三部分是连接前两种元素的布线要素既FPGA内部互联资源。Comment by Administrator: 参考文献

## FPGA基本组成图示

上述FPGA三部分主要组成具体包括以下内容：可编程输入输出及内嵌RAM、可编程逻辑单元、布线资源等。可编程逻辑单元负责逻辑功能的实现，布线资源则能够将各个单元进行链接，而内嵌RAM可配置为各种存储形式，输入输出负责FPGA与外部模块的通信，在最新及更高的FPGA上还会集成DSP、LUTRAM、CARRY等高级资源，如Xilinx最新的Virtex-7系列FPGA。而在以上FPGA的结构中，设计者及其支持软件中最关

系、最重要的是可编程逻辑单元，它主要包括触发器和查找表两部分。查找表在功能上等效为一种特殊的静态存储器SRAM。Comment by Administrator: 参考文献

可编程逻辑单元中，查找表表示的是1个字只有一位的内存表，字数取决于地址的位数。FPGA上查找表的存储单元大多使用SRAM实现。查找表的结构如图2.2所示

#### LUT结构图示

如图2.2中所示的3输入查找表，它可以实现任意3输入的逻辑函数。一般k输入的查找表由2k个SRAM单元和一个2k输入的数据选择器组成。查找表的输入就是内存表的地址信号，而输出就是该地址所选字的1位数据。K输入的查找表可以实现2<sup>2k</sup>种逻辑。

#### LUT基本工作原理

LUT基本工作原理如图2.3所示，上图左侧为电路的实现逻辑及真值表，上图右侧则对应的等效查找表。由图2.3可以知，查找表中的实际内容是将逻辑的输入等效转化为对应的存储地址，对应的输出的则是存储地址中的具体内容。最终的结果是查找表将对应的复杂逻辑运算对等的映射为存储器中的数值，这也是使用FPGA进行验证效率远远高于使用软件模拟的方法的主要因素。所以LUT资源是在对大型集成电路设计使用FPGA进行原型验证时的紧俏资源。此外，当面对的是时序逻辑电路，则FPGA会通过在每个查找表的后侧通过触发器的配置来完成。

#### D触发器

可编程逻辑单元中的另一种资源FF (Flip Flop) 触发器,其基本结构如图2.4所示，工作原理同数字电路中所使用的触发器原理一致，也称为双稳态门，或称双稳态触发器。当触发器接收到输入脉冲的到来时，触发器输出就会根据规则改变状态，然后保持这种状态直到收到另一个触发。触发器 (flip-flops) 电路之间是相互关联的，从而形成数字集成电路的逻辑门。触发器通常可用来存储一比特的数据。有几种不同类型的触发器 (flip-flops)，如T (切换)、S-R (设置重置) J-K (也可能称为Jack Kilby) 和D (延迟)。典型的触发器一般包括0个、1个或2个输入信号，以及clk时钟信号和输出信号。触发器资源通常在大型RTL设计进行高密度FPGA原型验证时也是一种常常面临资源紧俏的情况。

#### FPGA逻辑综合与技术映射

在本文中提到的FPGA原型验证方法流程中的起点如图2.5均为RTL级硬件描述语言代码，故本章我们先对硬件描述语言verilog HDL语法及逻辑进行简略的介绍，同时进一步介绍基于Verilog HDL的逻辑综合及工艺映射基本原理及基本算法，上述内容将为后续我们进一步建立可工程化的快速资源估算模型提供基础。

#### 行为级硬件描述语言

#### FPGA设计基本流程

Verilog HDL是用以描述硬件行为级的一种高级硬件描述语言，其表达形式通常以文本或字符形式来描述数字逻辑系统的结构和行为。通过合理的使用该语言可以表示正确表示逻辑电路图、逻辑表达式等，还可以表示数字逻辑系统所完成的逻辑功能。Verilog HDL是当前世界上最流行的两种硬件描述语言之一，都是在20世纪80年代中期开发出来的。前者由Gateway Design Automation公司（该公司于1989年被Cadence公司收购）开发，其依据的标准为IEEE标准。根据IEEE所描述的Verilog HDL是一种用来描述和创建电路系统所有阶段的表示形式，由于Verilog HDL是机器可读的、人类可读的，所以该语言能够支持对硬件设计进行开发、正确性验证、合成和测试、硬件设计数据的通信和维护、修改[16]。

在Verilog HDL中有两种基本语句：第一种是连续型赋值语句，在此表达式中寄存器和线网都可以用于连续驱动线网，从而辅助实现结构化的基本建模；另一种是过程型赋值语句，在此表达中将使用寄存器来存储寄存器与线网的运算结果，从而实现行为级的描述建模。一个完整的RTL级行为描述设计通常会包含多个模块，每个模块都将用于描述该模块对应的功能描述逻辑[16]。这些模块将被开发者进行抽象，然后通过线网连接。

而在典型FPGA设计流程中，或大型高密度FPGA原型验证流程中，首选通过使用Verilog HDL等硬件描述语言编写寄存器传输级（RTL）代码来描述电路设计的行为级逻辑即RTL级描述。RTL级描述通过逻辑综合得到电路网表（Netlist），通过对网表进行优化、技术映射、FPGA布局布线等操作后，最后生成FPGA配置数据（bitstream）。其中对于大型原型验证系统最为关注的前端工程就是逻辑综合和技术映射，这个过程是验证Verilog HDL语法正确性和计算该设计所需FPGA片上资源总数的重要过程。其中逻辑综合和技术映射的过程就是将Verilog HDL语言描述的寄存器级代码转换映射至具有工艺概念的逻辑器件的过程，也是本文中结合机器学习人工智能进行FPGA资源估算设计过程中重点关注过程。

#### FPGA逻辑综合

对于FPGA数字电路设计而言综合简而言之就是从采用硬件描述语言构造的寄存器传输级(RTL)电路抽象级模型构造出对应的门级电路网表的过程。

综合的过程就是将较高级的设计描述自动转化为较低层次描述的过程，综合分为行为综合和逻辑综合，行为综合是从算法表示、行为描述转换到寄存器传输级；逻辑综合是从RTL级描述转换到逻辑门级，包含触发器的过程，也是在资源计算中最关注的过程，还有版图综合和结构综合，是将逻辑门表示转换到版图表示的过成或转换到PLD器件的配置，版图综合和结构综合以及行为综合均不是我们主要关注的流程，后续将主要介绍逻辑综合这一流程。

综合整个ASIC设计流程的中间步骤，综合后生成的网表文件，其本质是基于导线来进行连接的RTL级功能块（如FF、LB和MUX等）所组成。而这里一般就需要使用RTL模块构造器来通过用户所注明或进一步指定的目标工艺元件库中构造或从已有库中获取必须的RTL级功能模块。综合产生网表文件后，逻辑优化器会读入网表，并且会按照用户配置的FPGA片上资源要求和时序约束为目标来对网表进行优化。并且，这些资源约束或称为面积约束和时序约束也能够指定模块构造器合理的选取或生成RTL级功能块。既然综合器能够将硬件描述语言转换成门级网表，那么综合器必定有一种映射的机制，能够实现Verilog代码中的某些变量、组件映射成对应的硬件元件，这种映射机制将是后续我们基于机器学习进行资源映射方法研究所重点关注的对象。

#### FPGA逻辑综合流程

①综合首先需要使用RTL代码构成的电路行为级描述进行编译使其转换为GATE级描述；

②RTL模块构造器从用户指定或综合器所携带的工艺库中里提取并生成对应的模块；

③逻辑优化器读取未优化的门级网表，并按照用户设置的面积和时序约束，进行优化；

由于本文中主要涉及计算RTL模块构造器从工艺库中提取对应的语法及模块进行Technology Mapping映射后所得到的资源，所以当我们聚焦于本文所述的基于机器学习资源估算模型，更加关注于Verilog HDL语句是如何转换成实际硬件器件的，如：数据类型是如何映射成硬件的，常量是如何转换成逻辑数值的，语句是如何转换成硬件的。下面我们将详细分析下综合中各个Verilog HDL语句及器件的综合原则。

#### 1、储值单元的综合原则

对于FPGA上的实际硬件主要有三种基本的储值单元分别是：wire、锁存器、触发器，对应于Verilog HDL语言中，变量可以是net类型也可以是reg类型，它们的综合原则分别是对于net类型的变量则只能总合成导线即wire类型，对于reg类型变量可能综合为导线即wire类型，rege类型的也可能被从合成锁存器或触发器，但是若reg类型的变量在一个always语句块中作为临时变量，则不会被总合成存储器。

#### 2、连续赋值语句的综合原则

##### 连续赋值语句综合图示

连续赋值语句（Continuous Assignment）语句对应的硬件实现如图2.7所示，从赋值语句等号的右边提取出逻辑，用于驱动赋值语句左边的线网即Net，连续赋值语句的逻辑功能实现后，赋值表达式中信号的变化随时会被传输至赋值线网型数据的取值上[16]。

#### 3、过程赋值语句的综合原则

过程过程赋值语句（Procejural Assignment）语句对应的FPGA的硬件实现是：从赋值语句也就是‘=’或‘<=’的右边提取出逻辑，用于驱动赋值语句左边的变量，但是有个语法前提，就是用于驱动赋值语句左边的变量是reg类型，且只有Always语句块中的过程赋值语句才可以被综合器综合，若出现在initial语句块中将被仅用于仿真。过程赋值语句有以下两种分别为：阻塞赋值语句（Blocking Assignment Statement）、非阻塞赋值语句(Non-blocking Assignment Statement)

##### 阻塞赋值语句综合

##### 阻塞赋值语句综合图例

阻塞赋值语句简而言之就是在一个always语句块中，语句是从上到下顺序执行，综合时将会在依据赋值语句逻辑原理结合always块的触发器来实现逻辑综合，阻塞赋值语句RTL级描述与电路映射如图2.8所示。

##### 非阻塞赋值语句综合



## 非阻塞赋值语句综合图例

非阻塞赋值语句简而言之是在一个always块中，语句是并行执行的，非阻塞赋值语句RTL级描述与电路映射图2.9所示，对比上图可以得到对于一个Always块中的阻塞或非阻塞语句的综合最大的区别在于是并行还是顺序执行，但是对于综合后器件的数量，也就是我们所关心的资源数量并没有明显的影响，主要影响使用资源数量的关键在于赋值语句右边的驱动逻辑的组成。

## 4、逻辑运算的综合原则

Verilog HDL语句中的逻辑运算符将会被直接综合成对应的基本逻辑门,如与或非、异或门等。

## 非阻塞赋值语句综合图例

## 5、算术运算符的综合原则：

在Verilog HDL语句中支持各种算术运算符如+、-、\*、/等，在进行综合时RTL构造器会将其构造为加法器、减法器、除法器 etc 源语器件。如图2.11所示为加法运算综合后器件映射图。

## 算术运算符综合图例

## always语句块的综合原则

在Verilog HDL设计中用的最多的就是always语句，always语法既能用于描述组合逻辑也能描述时序逻辑，主要区别在于敏感触发信号是否为时序信号，always语句块中多包裹其余语法使用，单独使用不具有任何意义。

## if及条件语句的综合原则

在Verilog HDL语句中的单条if-then语句常表达条件判断行为，故常被综合为MUX数据复选器，若为嵌套组合预计，常会被逻辑综合为售条件控制的纯组合逻辑电路，如图2.12所示。

## if语句综合图例

## case语句的综合原则

在Verilog HDL语句种，case包含两类条件表达式和选择项表达式，case语句可以看作是嵌套的if语句，所以其综合原则与嵌套if条件语句类似，如图2.13所示。

## case语句综合图例

## 10、循环语句的综合

对于循环语句，如果循环内部的赋值语句之间不存在相互依赖性，则这些语句可以展开为并行执行的语句，且语句执行先后顺序不会影响到最终的结果，For循环语句在逻辑综合后会依据循环体内部的逻辑构成级联的链式组合逻辑电路。

## 循环语句综合图例

## 11、函数语句的综合

在verilog HDL语句中，函数代表了纯组合逻辑。函数在综合的时候，被展开成了内联的代码。任何在函数中定义的局部变量都被当做纯粹的临时变量，仅被总合成导线，且由于函数没有时序控制，因此综合结果为组合逻辑，如图2.15所示。

## 函数语句综合图例

## FPGA技术映射

在综合优化过程中，技术映射是将技术映射作为综合优化的重要组成部分，也是FPGA支持软件工具EDA中最为关键的部分，技术映射是面向FPGA的EDA软件中至关重要的一步，其映射和优化结果对整个FPGA设计流程中下一阶段的布局布线等步骤有着决定性的作用，进而最终影响芯片的性能成本，故而已形成了以面积优化、时间延迟优化、功率优化以及面积和延迟同时优化为目标的一系列研究成果。



工艺映射是指将不依赖于任何工艺的网级网表转换为由特定FPGA逻辑单元所表示的网表的过程。这里所说的逻辑单元依赖于特定的FPGA架构，是由查找表或MUX等逻辑电路实现的FPGA上的最小逻辑单位。工艺映射是从HDL开始的逻辑转换的最后一步，因此对最终电路实现的质量(面积、速度、功耗等)至关重要。

#### 工艺映射流程图

技术映射实际上就是将电路从布尔网络到实际实现的电路的一种映射过程。其目的是对用户输入的设计电路经过逻辑综合后生成的网级网表来通过先进的算法来进行划分，使得原设计中的每一小块电路的逻辑功能可以借由FPGA的一部分CLB来实现。我们可以参考图2.16来进一步对技术映射的流程进行分析。

由图2.16可以发现工艺映射的流程是完成网级网表的输入至实际工艺下可实现的工艺网表输出这一个过程，网级网表我们可以理解为由一组逻辑表达式构成的布尔网络，其每个节点可以是任意的逻辑功能组合单元。

由图2.16可以看出划分是工艺映射流程的第一个步骤，划分实际上是对布尔网络进行功能上的划分，它将多输出的网络转化为一组单输出的子网来简化映射过程。划分完成以后，原电路可由一组单输出布尔子网络表示。根据工艺映射算法的不同需要原电路可由有向无环图或一组树来表示。

逻辑分解是将电路进行逻辑上的变化，目的是将复杂的逻辑单元用基本函数的形式（如两输入的与门、与非门、或门，单输入的非门）表达，以便于以后的匹配和映射。

经过逻辑分解后的布尔网络由各节点组成且每个节点都只有基本函数关系。经过划分和逻辑分解之后下一步就是匹配。之前的划分和分解都是对布尔网络进行与工艺无关的处理，而匹配则是要把布尔网络和实际的单元库相联系。就是用实际工艺下可实现的单元库对布尔图进行匹配也就是把理想的布尔网络用库中的实际单元实现的过程。工艺库中的实际单元要实现划分的子网络的功能。匹配还可以分为结构匹配和布尔匹配。结构匹配是利用结构的特点来进行匹配的。在布尔式进行代数分解后关系式可以用树或图来表示这样表达式之间的匹配就可以转化为图之间的同构验证。匹配问题的规模则受限于库单元输入数的最大值。布尔匹配可以利用逻辑函数对无关项进行操作也就可以考虑所有匹配。但是由于考虑到时间和速率大多数的商用工具使用较多的依然还是基于树或图的结构匹配算法。

工艺映射自动化过程的最后一步是覆盖，它的作用是根据实际电路的要求，在所有节点的多种可行匹配集中选出足够的匹配来覆盖整个网络的所有节点。也就是依据一定的规则分隔出部分网表进行匹配，并从匹配集中选出库单元，完成布尔网表到工艺网表的实现。对每一个选出的匹配，要保证每个单元库的输入和其它匹配库单元的输出相连，即覆盖完成后，整个网络都被匹配库单元覆盖而且匹配库单元间又不互相重叠。某一部分网表可能有许多库单元与之匹配，但是现实采用那一个要根据具体情况，从可完成覆盖的各组匹配中选出优化目标面积，深度，功耗等最优的一组匹配。覆盖的核心优化算法是如何分割布尔网表、如何选取匹配集。最后以实际网表的形式输出。

下面我们通过分析具有代表性的FlowMap的LUT工艺映射原理，来进一步理解上述工艺映射流程。

在FPGA中的基本资源中查找表的输入数是既定的，查找表能实现输入数不大于自己的任意逻辑，而FPGA的设计就是要从目标电路的逻辑函数中不断分离出既定输入数之内的逻辑，并将其映射到查找表上。工艺映射包含多种工艺映射算法，既将目标电路网表转换到K输入的查找表(k-Lut)的工艺映射过程由下面两个步骤组成：

- 1)分解：网级网表实际上都是以布尔网络的形式来表示的。现将布尔网络的各个节点不断分解，直至输入数小于查找表的输入数K。
- 2)覆盖：基于过程1得到的布尔网络，使用某种基准对输入进行切分，使用k-Lut覆盖多个节点，覆盖的过程是一种单位时间内找到逻辑层数最优解的办法；

我们将通过以下实例对工艺映射原理进行进一步说明。

首先将图2.18中的网级网表转换为图中的DAG有向图的方式来表示。最上层的节点成为主输入，最下层的节点称为主输出，从而可以得到如图2.19所示的映射范围，图2.20所示为在映射范围下进行切分和标注求解的过程及将映射范围的有向图依据主输入及主输出进行逻辑分层，从而标注拓扑序列，进而完成覆盖。其具体步骤如图2.17所示。

Mapping算法伪代码

电设计转化为DAG图

逻辑划分





## 器件覆盖

通过上述计算可以得到每一层的标注为依据上层计算而来的最小值，因此可以保证使用最少的逻辑层数。之后对电路的主输出开始进行器件映射，每映射一个后主输出将发生变更，对所有的主输出依次进行算法映射后，便可得到相对应的查找表及网表图，如图2.20所示。

在对目标电路进行切分和映射的过程中，通过改良评估函数可以实现各种工艺映射算法，如：哥伦比亚大学的Steve Wilton团队开发了改善功耗的EMap，多伦多大学的Stephen Brown开发了IMAP算法等。

## 机器学习资源估计方法与模型

本节将主要介绍为建立基于机器学习的FPGA设计资源估算的可行性与模型的理论基础，包括机器学习与EDA想结合的可行性、随机森林，神经网络（ANN）及模型评估指标。

### 机器学习与EDA结合的可行性

机器学习是一门多学科交叉专业，涵盖概率论知识，统计学知识，近似理论知识和复杂算法知识，使用计算机作为工具并致力于真实实时的模拟人类学习方式，并将现有内容进行知识结构划分来有效提高学习效率[30]。

(1) 机器学习是人工智能的科学，该领域的主要研究对象是人工智能，特别是如何在经验学习中改善具体算法的性能[30]。

(2) 机器学习是对能通过经验自动改进的计算机算法的研究。

(3) 机器学习是用数据或以往的经验，以此优化计算机程序的性能标准。

通过机器学习的定义同时结合FPGA设计逻辑综合流程及技术映射原理，我们可以得出对于大型FPGA设计在综合阶段存在的资源估算的时间长的弊端，我们可以使用机器学习方式以Verilog HDL的各种语法及参数的数量及值作为特征，Vivado综合的资源数值为标签，来训练监督学习的回归模型，从而实现快速的资源估算。基于机器学习实现资源估计方法的流程如图

### 机器学习实现RTL级设计资源估算流程

#### 特征提取方法

对于机器学习或人工智能而言，模型的选择与好坏并不能最终决定预测精度的上限与下限，最终模型的上限取决于特征与样本的优劣，所以特征对于机器学习模型的学习与预测是相当重要的。

特征的选择是一个相当重要的数据预处理的过程，所以在选择学习模型与训练器前需要先对样本进行特征提取，对于在RTL级进行资源预测这个命题，我们需要在RTL级提取尽可能多的语法特征，从而确保不会丢失重要特征，虽然会获得很多无关特征或冗余特征，但在进行特征工程及训练模型迭代时可以进一步对特征进行优化，从而保留相关特征和重要特征。

由于RTL级设计是基于Verilog HDL语言开发的编程语言，所以我们采用尽可能细致的特征提取方法，即将Verilog HDL抽象为数据结构，并结合IEEE 1800所示Verilog语法标准，通过进一步解析数据结构来尽可能全面的提取Verilog HDL语言的语法特征，从而构建在基于2.2节原理的基础上的合理特征集。具体的特征提取算法与特征提取的构建将在第四章进行详尽的阐述。

## 随机森林

在机器学习中，随机森林是通过包含多个决策树若学习器进而构成的强学习器。随机森林是Leo Breiman和Adele Cutler发展出推论出随机森林的算法，这个术语是1995年由贝尔实验室的Tin Kam Ho所提出的随机决策森林（random decision forests）而来的。这个方法则是结合Breimans的"Bootstrap aggregating"想法和Ho的"random subspace method"以建造决策树的集合[30]。

### (1) Bagging

Bagging是并行式集成学习方法中比较具有代表性的学习方法，它基于自主采样法，即给定一个数据集拥有固定m个数据样本，学习算法先随机拿出一个样本放入采样集中，再通过将该出去的样本放回数据集，使得在每一次采样时均有可能取到之前的样本，这样经过覆盖所有样本的次数的随机采样，最终会得到采样集包含m个样本，初始训练集中有的样本在采样集中多次出现，有的则从未出现。由式（2-1）可知，初始训练集中约有63.2%的样本出现在采样集中。

### (2-1)



通过上述算法可以采样出个含个训练样本的采样数据集，随后基于每个采样集训练出对应的弱学习器，再通过将这些基学习器的学习结果进行进一步的结合。从而实现Bagging算法的流程，若基学习器的计算复杂度为 $T$ ，则通过采样方法可知该算法的复杂度约为 $T$ ，考虑到采样与投票过程的复杂度很小，而通常是较大样本数据而言是一个较小的常数，因此，训练使用Bagging集成的学习器与使用一个弱学习算法训练所需要的时间复杂度同阶，从而证明Bagging是具有较低时间复杂度、较高学习效率的集成学习算法。

从偏差-方差分解的角度看，由于Bagging学习算法的主要关注点在于如何降低方差，因此它在易受样本扰动的学习器上的优秀表现更为明显。Bagging的算法伪代码如图2.21所示。

Bagging算法伪代码

决策树

决策树是一种常见的机器学习算法，我们以二分类任务为例，希望从给定训练数据集学得一个模型用以对新示例进行分类，决策树学习的目的是为了产生一棵泛化能力强，即处理未见示例能力强的决策树能够对泛化样本进行正确且合理的决策，其基本流程遵循简单且直观的“分而治之”策略，决策树学习基本算法伪代码实现如图2.22所示。

决策树学习基本算法

显然决策树的生成是一个递归的过程。在决策树的决策过程中我们更关心的肯定是如何划分，或者说如何选择最优划分属性，即结点的“纯度”，决策树中衡量一个结点的纯度的高低有以下几类指标：

信息熵公式：

(2-2)

增益率公式：

(2-3)

基尼指数：

(2-4)

随机森林

随机森林是Bagging的一个扩展变体，是一种由多个弱学习器（决策树所集成的强学习器）。RF在以决策树为基学习器构建Bagging集成的基础上，进一步在决策树的训练过程中引入了随机属性选择。具体来说，传统决策树在选择划分属性时是在当前结点的属性集合（假定有个属性）中选择一个最优属性；而在RF中，对基决策树的每个结点，先从该结点的属性的集合中随机选择一个包含个属性的子集，然后再从这个子集中选择一个最优属性用于划分。

随机森林在许多显示任务重展现出强大的性能，随机森林对Bagging做了很小的改动，但是继承了Bagging中基学习器所具有的多样性，仅通过样本扰动而来不同，随机森林中基学习器的多样性不仅来自样本扰动，还来自属性扰动，这就使得最终集成的泛化性能可通过个体学习期之间差异度的增加而进一步提升。

通过对随机森林算法进行研究，发现随机森林相较于传统的机器学习模型具有训练速度快、预测效果好、泛化能力强、鲁棒性强等优势，可有效避免模型过拟合的现象出现，适用于高维数据的处理。

随机森林的回归预测结果是棵决策树的预测结果取均值而来的，它满足以下定理：

定理1 当时，

(2-5)

式(2-5)右侧部分表示随机森林的泛化误差，将其记为 $\sigma$ 。则表示一棵决策树的平均泛化误差即

(2-6)



定理2 对所有随机生成的训练集有,

(2-7)

式(2-7)中是在与相互独立的情况下, 残差和的加权相关系数。

上述定理1和定理2给定了使随机森林精确的前提是: 残差间的相关系数低以及错误决策树数目较少。为降低决策树的平均误差, 随机森林回归选择对相关系数进行加权处理。随机森林的具体算法步骤伪代码可概括为下图2.23所示。

随机森林模型算法步骤

神经网络

神经网络是深度学习的核心。其在当代用途广泛、功能强大且可扩展, 使其非常适合处理大型和高度复杂的机器学习任务, 例如对数十亿张图像进行分类, 例如Googal的Images, 或者为语音识别服务提供更支持比如苹果的Siri等。下面将主要介绍神经网络中的感知器计算原理及多层感知机。

感知器

神经网络是深度学习的核心。其在当代用途广泛、功能强大且可扩展, 使其非常适合处理大型和高度复杂的机器学习任务, 例如对数十亿张图像进行分类, 例如Googal的Images, 或者为语音识别服务提供更支持比如苹果的Siri等。

神经元计算图示

感知器是最简单的ANN架构之一。它基于稍微不同的人工神经元如上图所示, 称其为阈值逻辑单元 (TLU), 也称为阈值逻辑值单元 (LTU)。输入和输出是数字 (而非二进制开关值), 并且每个输入连接都与权重相关联。TLU计算其输入的加权和数学公式, 然后阶跃函数应用于该和并输出结果, 感知器公式xxx为最常用的阶跃函数, Heaviside阶跃函数。

(2-8)

(2-9)

感知器仅由单层TLU组成, 每个TLU连接到所有的输入。当一层中的所有神经元都连接到上一层中的每个神经元时, 该层称为全连接层或密集层。感知器的输入被送到称为输入神经元的特殊直通神经元: 它们输出被送入的任何输入。所有输入神经元形成输入层。此外, 通常会添加一个额外的偏置特征 ( $x_0=1$ ): 通常使用一种称为偏置神经元的特殊类型的神经元来表示该特征, 该神经元始终输出1。具有两个输入和三个输出的感知器如图2.25所示。该感知器可以将实力同时分为三个不同的二进制类, 使其称为多输出分类器。

感知器结构图示

借助线性代数的数学关系, 通过以下公式我们可以同时为多个实例高效地计算出一层人工神经元的输出。计算全连接层的输出

(2-10)

关于感知器的训练, 感知器的训练算法在很大程度上受Hebb规则的启发, 当一个生物神经元经常出发另一个神经元时, 这两个神经元之间的联系就会增强, 也就是说, 两个神经元同时触发时, 它们之间的连接权重会增加。更具体的说, 感知器一次被送入一个训练实例, 并且针对每个实例进行预测。对于产生错误预测的每个输出神经元, 它会增强来自输入的连接权重, 这些权重将有助于正确的预测如公式 (2-11) 所示。

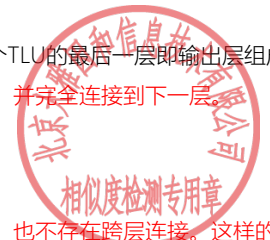
(2-11)

多层感知机

多层感知器也称人工神经网络, 则是在上述感知器的基础上, 由一层输入层、一层或多层TLU即隐藏层和一个TLU的最后一层即输出层组成。靠近输入层的层通常称为较低层, 靠近输出层的层通常为较高层。除输出层外的每一层都包含一个偏置神经元, 并完全连接到下一层。

多层感知机结构图示

常见的神经网络是如上图所示的层级结构, 每层神经元与下一层神经元全互联, 神经元之间不存在同层连接, 也不存在跨层连接。这样的神经网络



神经网络结构通常称为“多层前馈神经网络”，即多层感知机，其中输入层神经元接受外界输入，隐藏层与输出层神经元对信号进行加工，最终结果由输出层神经元输出；换言之，输入层神经元仅是接受输入，不进行函数处理，隐藏层与输出层包含功能神经元。神经网络的学习过程，就是根据训练数据来调整神经元之间的“连接权”以及每个功能神经元的阈值；换言之，神经网络所“学”到的东西，蕴含在连接权与阈值中。

当一个ANN包含一个深层的隐藏层时，就属于深度神经网络（DNN），其分为正向传播和反向传播，反向传播是一种沿用至今的训练算法，其使用的是有效的技术自动计算梯度下降，在仅两次通过网络的过程中，即一次前向，一次反向，反向传播算法能够针对每个模型参数计算网络误差的梯度。换言之，它可以找出应如何调整每个连接权重和每个偏置项以减少误差。一旦获得了这些梯度，它便会执行常规的梯度下降步骤，然后重复整个过程，知道网络收敛到解。

#### 典型多层感知机架构

#### 机器学习性能评估指标

评价模型性能的指标有很多种,需要我们根据特定的任务选择合适的评价指标。不恰当的评价指标不仅无法模型进行评估,反而会造成误导,得出错误的结论。一个评价指标可以展现模型在某一方面的能力,所以一般情况下,需要构造多个评价指标,来对模型进行全面的分析。针对大型RTL设计的FPGA资源估算模型的需求分析,其为回归问题模型,对于回归问题模型的特点有如下几种模型性能评估指标:

##### 均方误差

均方误差是在回归问题的机器学习模型中常见的泛化误差衡量指标，均方误差是反映估计量与被估计量之间差异程度的一种度量。

均方误差是评价点估计的最一般的标准，自然，我们希望估计的均方误差越小越好，均方误差公式如(2-12)所示。

(2-12)

式(2-11)说明，均方误差由点估计的方差与偏差的平方两部分组成。

##### 均方根误差

均方根误差，也称做标准误差，均方根误差是预测值与真实值偏差的平方与采样次数 $n$ 比的平方根，在实际测量中，观测次数总是有限的，真值只能用最可信赖（最佳）值来代替。标准误差对一组测量中的特大或特小误差反映非常敏感，所以，标准误差能够很好地反映出测量的精密度。这正是标准误差在工程测量中广泛被采用的原因。因此，标准差是用来衡量一组数自身的离散程度，而均方根误差是用来衡量观测值同真值之间的偏差，它们的研究对象和研究目的不同，其计算过程公式如(2-13)所示。

(2-13)

#### 3、R-Squared

是统计学里常用的统计量，的名字是coefficient of determination，另一个名字是Nash-Sutcliffe model efficiency coefficient。给定一系列真值和对应的预测值，的定义如公式(2-13)所示。

(2-14)

的含义是，预测值解释了变量的方差的多大比例，衡量的是预测值对于真值的拟合好坏程度。通俗理解，假定的方差为1个单位，则表示使用该模型之后，的残差的方差减少了多少。比如等于0.8，则使用该模型之后残差的方差为原始值方差的20%。

：最理想情况，所有的预测值等于真值。

：一种可能情况是“简单预测所有值等于平均值”，即所有都等于真实值的平均数。

：模型预测能力差，比“简单预测所有值等于平均值”的效果还差。这表示可能用了错误模型，或者模型假设不合理。

的最小值没有下限，因为预测可以任意程度的差。因此，其范围是。

#### 快速资源估算模型

本章将结合对第二章所述的FPGA逻辑综合原理与技术映射原理的研究，使用C++语言设计并构建基于传统框架下的快速资源估算模型，并



基于该模型结合第一章所述高密度FPGA原型验证系统所需要解决的关键问题及基本需求增加算法优化与约束项优化，实现符合基本原理与能够解决实际工程问题的快速RTL级设计资源估算模型。

#### Verific及相关数据结构

##### Verific平台介绍

Verific Design Automation是Verific公司开发的一款用于解析与构建SystemVerilog、VHDL和UPF的解析器平台，使其客户能够以低成本快速开发高级EDA产品。

Verific分析创建解析树并执行类型推断以解析标识符的含义。解析器分析器模块支持整个系统Verilog IEEE 1800（包括Verilog IEEE 1364）、VHDL-IEEE-1076和UPF-IEEE-1801语言。生成的解析树数据模型附带了大量成熟的C++ API。Verific的解析器平台分发给C++源代码，并在所有32位和64位UNIX、Linux、MAC和Windows操作系统上进行编译。Verific的解析器平台目前在全球许多公司的生产和开发中使用，从EDA初创公司到《财富》500强半导体供应商等。

我们将使用Verific，并基于Verific生成的语法树来完成后续模型构建操作，由于Verific具有当前行业认可的Parse和Analyze功能，并基于C++平台实现开发，提供广泛的语法树API作为二次开发的基础，故我们选择在Verific的基础上实现可商业化使用的快速资源估算模型。首先需先对Verific所构造的基本数据结构进行相关了解及分析，从而选择构造合理的数据机构及API进行模型建模实现；Verific主要数据结构分为以下两个主要部分：基于RTL的语法树结构和基于Netlist的Database结构，下文将主要介绍构建资源估算模型使用的主要数据结构。

##### Parse-Tree数据结构

Verific在RTL级别Verilog VHDL的语法树Parser上有丰富的经验，通过对语法进行文本级别的流解析在内存中构建层次树（Hierarchy-Tree）结构用以表述Verilog VHDL语法信息，层次结构树（hier Tree）是设计层次结构的独立于语言的表示形式。设计的层次结构由Verilog语言中使用的模块实例化和生成语句定义。随后在本文档中，我们将此数据结构称为hier树或Parse-Tree。这个层次结构树结构是一个轻量级的静态详细层，位于分析的解析树之上，保存着关于各种设计对象的关键信息。树的每个节点都存储有关它所表示的设计对象的信息，以及该层次结构路径中特定于该设计对象的参数和泛型等常量的值。语法树结构主要由以下几种类结构：

①VeriTreeNode是所有Verilog树节点元素的基类。因此，它包含行文件信息和在所有解析树节点上的数据信息。其中VeriExpression、VeriIdDef、VeriStatement等中非常常用的派生类，用于对Verilog Design作进一步数据构造，下图为VeriTreeNode的继承示图：

##### VeriTreeNode类视图

②VeriIdDef是标识符定义树节点，是标识符节点的基类。Verilog中声明的任何标识符都将成为VeriIdDef结构类。下图为VeriIdDef的类关系视图，其中VeriIdDef的继承类VeriModuleID、VeriOperatorID、VeriInstID等为进一步对Verilog语言语法的描述类。

③VeriExpression是所有Verilog表达式解析树节点的基类。其继承类用于描述包括常量、名称引用、运算符以及出现在实例化端口关联列表和模块端口列表中的端口关联信息等。下图为VeriExpression的类关系视图。

④VeriStatement是所有Verilog statement 节点的基类，包括所有Always语句块等语法

⑤VeriModule位于任何Verilog解析树的顶部，是模块节点的基类。它以解析树的形式表示一个完整的Verilog模块。VeriModules存储在静态哈希表中，该哈希表表示已分析的所有Verilog模块。

##### VeriModule类视图

上述5种数据结构为Verific中最基本的基类，所有语法树的语法表述及存储均可由其中一种进行表述，在本文中所有RTL代码的语法将会使用上述类进行语法树处理及构造，从而获取正确的设计层级及内容。

##### Netlist数据结构

由于传统技术映射是在网表级的基础上通过器件库使用图划分算法，将网表节点映射至对应的器件，所以为构建基于传统方法的快速资源估算模型，也需要使用Verific的Netlist数据结构来实现，Verific Netlist Database数据结构如下：

##### Verific-Netlist-DataBase数据结构图



①Libset是一个单例对象（只实例化一次的对象），它包含（char\*）lib\_name到Library\*映射的映射。Libset没有所有者，因此被视为全局对象；

②Library是Libset拥有的对象，它包含所有Module模块命名到cell指针映射的映射。库对应于它的VHDL或Verilog等价物。例如，在VHDL中，将用户实体体系结构对分析到一个名为“user”的库中，然后详细描述解析树，那么将创建一个名为“user”的库对象并将其添加到全局库集中。

③Cell是库所拥有的对象，它包含Netlist\_name到netlist\*映射的映射。当处理Verilog时，单元是包含单个网络列表的模块。就VHDL而言，单元是一个可以有多种体系结构（网络表）的实体。

④网络列表是数据库层次结构中的关键组件。它是一个由单元拥有的对象，包含其他网络列表实例化形式的功能（通过称为网络、端口和端口引用的对象连接在一起）。它还包含一组自身正在实例化的引用（从其他网络列表中）。

如上图所示，基本数据结构为LibSet用以存储所有的Library库，Library库为设计网表存储基本数据结构，Cell结构对应为VHDL代码中Module模块，每个Cell尤其仅有一个对应的Netlist结构，Netlist结构用以存储该Module所有互联信息及实例化信息，如：Port、Net、Instance，包括Net位宽、方向等。

基于Verific的上述数据结构以及相关的API，使得我们可以在后续基于C++语言实现传统快速资源估算模型提供基本数据基础。

## 模型构建

本节将主要介绍使用C++构建基于传统框架下的快速资源估算模型的算法分析及具体实现步骤。

## 模型流程分析

基于第二章节所述逻辑综合及资源映射原理，我们对模型的构建做出进一步的分析，模型输入为RTL级设计文件即Verilog VHDL文件，模型输出为该设计所需FPGA资源数目，其中对于面向大型RTL设计Partition中**所需的最重要的关键的资源为FF及LUT**，文件流如下图所示。

## 快速资源估算模型文件流

基于Technology Mapping的原理进一步分析，为实现传统框架下较为准确的资源估算，则需要将输入的原始设计从RTL级代码转换至Netlist级别才能实现进一步的Technology Mapping，但是由于获取最直接的网表解构就需要使用综合器对RTL设计进行综合，通过前面对综合原理的分析，可知道综合主要分为以下几步，即RTL设计细化，Flatten，综合优化，综合约束，技术映射等几个步骤，但由于需要面向大型FPGA设计，综合器耗时太长，所以我们选择通过使用Verific来对设计进行静态细化及初步细化，使RTL级设计在内存中转变为伪网表，通过在内存中对伪网表进行Flatten操作，将伪网表所有的层级结构UnGroup至顶层，从而实现与基本综合器相对应的操作的快速版本，对Flatten至顶层的设计依据TechMapping原理使用已经获得的器件库进行器件匹配，基于本节分析及初步设计结果，着手使用Verific对传统资源估算进行进一步分析及模型构建，具体流程如图所示，关键操作及相关原理将在下文着重分析。

## 快速资源估算模型流程图示

## 建模实现

我们以上一节中对快速资源估算模型分析的流程进行进一步的分析，同时进行实现：

### 1、Parse-Design操作

通过将原始设计使用Verific的Parser接口进行设计读入，由于需要商业使用，需要考虑多种用户输入情况，首先考虑常见的通过Filelist进行读入，并支持对于文件正确性校验及报错，并支持输入文件包括.sv.v.vh.svh等多种文件输入，同时支持通过路径进行设计输入，并对设计进行正确性校验，如图3.6所示。

## 快速资源估算模型Parse-Design流程

### 2、Elaborate操作

通过对上一步骤获得的完整输入设计的Parse-Tree进行Elaborate，从而获得对应设计的由Verific GTec Cell构成的伪Netlist的设计文件，Elaborate后的设计语法依旧满足Verilog HDL的语法信息，RTL Elaborate会使用门级的网表来表示之的RTL级代码。Elaborate会将RTL级代码进行逻辑展开，获得具有符合电路逻辑的门级视图网络和门级逻辑电路的VerilogHDL表述，该步骤为FPGA设计流程中综合的前置步骤，为下一





步进行优化的技术映射做好基础工作，Elaborate操作如图3.10所示。

RTL Elaborate操作图示

### 3、Flatten操作

在步骤2的基础上，我们将对设计进行Flatten操作，Flatten操作顾名思义是将现有包含层级关系的层级设计进行展平操作，也可以称为ungroup操作，参考下图样例可以更好的说明Flatten操作对于设计的操作，如图3.8所示。

Flatten操作电路图示

Flatten操作树结构图示

在FPGA设计综合流程中Flatten操作会将一些层次树底层的Module、逻辑提升至顶层，将可以合并的逻辑进行合并，同时将一些冗余的互联Net进行优化，为进步一的综合优化和器件映射做好准备，因为对于FPGA或者数字集成电路而言，它是没有层次的概念的，所以Flatten操作可以更好的让寄存器级设计与物理器件之间建立更进一步的联系。

### 4、基本器件库

由于我们中所设计的快速资源估算模型是面向Xilinx的FPGA的模型，所使用的基本器件库是Xilinx公司Virtex系列器件库，其中主要包含的为基本逻辑单元即CLB，具体为MUX、LUT1~6、各形式的触发器即FF、FDR等，一些IP器件由于知识产权等问题无法直接使用，如图3.10以LUT、FF、MUX为例进行说明。

RTL级基本器件库

### 5、Technology Mapping

对步骤4中完成Flatten后的网表文件进行TechnologyMapping，具体原理如2.2.3节中所述，具体操作为：将Flatten后的Netlist网表文件依据设计逻辑对将步骤四中的器件库的器件进行匹配和覆盖使用算法为EMAP，匹配后的网表将不存在基本的门级逻辑（布尔逻辑），而将会被器件库中的基本逻辑器件所替代，如MUX、D-FLIP-FLOP触发器、4-输入LUT器件等。

### 6、估算结果展示

下图为以上述思路构建的快速资源估算模型的初步估算结果，如图图3.11可以获得输入的RTL设计所使用的FPGA片上LUT1-6、BUFG、FF等器件资源数量。

快速资源估算模型计算结果

模型优化

通过图3.11与Vivado综合后结果对比，可以发现，在无优化的情况下仅对设计进行细化及基本技术映射虽然可以得到基本的资源，且在速度上得到了有效的提升，但是存在较大的误差，通过分析Xilinx的综合器Vivado细节及TechMapping，我们可以知道在与FPGA板载器件进行基本逻辑单元及触发器映射时存在优化项，可以理解为约束项，如：面积约束、功耗约束、时序约束、逻辑门约束等多种约束；结合面向大型RTL设计分割流程的实际需求，增加以下优化方案。

BottomToUP算法

由于使用Verific进行Elaborate处理及Flatten后无法得到每个层级对应的资源数量，且对整个设计进行Elaborate及Flatten所花费的时间较高，同时由于该资源估算模型面向RTLPartition进行使用，所以需要每个层级每个Inst的资源用于确认RTLPartition时正确的分割边界，所以我们从Verilog HDL与FPGA片上资源映射关系角度出发，在不考虑优化的前提下，每个Module的资源是由其所例化的Instance对应的Module的资源与该Module本身的胶水逻辑所组成，如下图样例所示，红框标注分别为例化Instance与胶水逻辑。

RTL代码中的Glue Logic和Instance

以上述分析为出发点，我们希望可以结合Verilog HDL设计建模方式，以该方式为参考，能够更加准确，更加快速的计算每个Module、Instance的资源，我们以RTL设计及计算资源所需数据为出发，设计以下数据机构用于存储每个节点的资源，如图3.13所示。





## BottomToUp算法数据结构

通过以上数据结构，用以实现Bottom to up的计算方式，通过树遍历算法及递归算法构造存储当前Parse\_tree对应的树结构及查找表，该查找表用以存储已被计算过的Module的资源，算法伪代码如图3.14所示。

### BottomToUp算法伪代码

通过上图所示算法，实现快速层次资源估算，输入为RTL级Verilog HDL设计，输出为该设计每个层级所使用的资源数量及相对应的树结构层次，若在输入时提供FPGA型号和资源，程序也会评估该设计所需要的FPGA的数量。如图3.15所示。

### 优化后快速资源估算模型输出

图中为开源设计openisc经过该快速资源估算模型新增的Bottom-to-Up算法计算所获得资源数量与每个Hierarchy对应的资源数量。

### 冗余逻辑优化及合并

在逻辑综合和技术映射时，综合开发环境会对电路规模、速度进行优化。综合的优化过程中会去掉常数输入、悬空输出等信号，还会针对源文件中未连接的输入或具有多个驱动的输出等问题报告错误或者警告。有时即使设计上有缺陷，工具也会按照既定的方式自行解决，但所生成的电路可能并不符合开发者的本意，所以调试时需要对信息特别留意。综合优化的过程还会对逻辑进行组合置换，因此需要注意的是，并非源文件中所有模块、寄存器、信号都会出现在综合结果的网表中。

通过前述步骤得到的快速资源估算模型，虽然已经可以得到资源的结果，同时若使用Bottom-to-Up的算法可以得到每个层级的资源结果，但是我们通过第二章的综合和技术映射的原理可以得知，在实际的综合和技术映射过程中，是对设计进行了大量的优化和处理的，所以如果我们不对当前的快速资源估算模型进行优化和约束，那么该模型的结果是不可用，没有价值和意义的。

所以我们为使资源估算的精确度更加高，所以需要符合逻辑综合和优化的过程，通过分析逻辑综合流程，在逻辑综合的工程中会对冗余逻辑进行优化，比如一些复选器或单纯的数值传递等非有效逻辑进行合并。

### 最小面积优化方法

通过本文第二章中对逻辑综合过程中优化步骤的分析，我们知道在综合开发环境会对RTL代码进行很多优化，从而提高设计的性能、减少硬件资源消耗等，所以在快速资源估算模型中也需加入相关优化项，来对RTL设计进行进一步的优化，从而使得资源估算模型的结果更加的精确。

首先我们考虑资源共享，即在互斥条件下共享算术逻辑单元，主要有关系运算、加法运算、减法运算、除法运算等。

### LUT-5 Base FPGA结构图示

根据FPGA的结构，可以知道不同系列、不同世代、不同型号的FPGA在逻辑单元的组成上是有很大的区别的，比如对于赛灵思的VU440，其基本逻辑单元即CLB单元是以LUT5为基础，同理也有以LUT1、LUT2、LUT3、LUT4、LUT6为基础，支持至最高为LUT8的FPGA，由于LUT1~8之间存在换算关系，LUT的本质就是如图3.17的2进制查找表，故而2个LUT1可以合并组成为一个LUT2，同理以LUT5为Base的FPGA，可以使用多个LUT5来组成LUT6等高维度资源，从而有效的是缩小FPGA布线后使用的面积即器件使用的数量。

### 4输入查找表真值表

在进行资源共享优化时我们遵循以下原则：

运算符和操作数均相同，则一定可以贡献逻辑单元。

相同的运算符，并且仅有一个共同的操作数，那么通过使用一个多路开关可以实现共享逻辑单元。

相同运算符，完全不同的操作数，可以共享逻辑单元。

不同的运算符，相同的操作数，可以共享逻辑单元。

不同的运算符，仅有一个公共操作数，可以通过引入多路开关来实现共享逻辑单元。

不同运算符、不同操作数，可以通过引入两个多路开关，从而实现逻辑单元的共享。



除去资源共享优化外，本文基于对RTL代码的分析，**增加公共子表达式优化**，即保存公共子表达式的值，在其后引用该表达式的地方可以直接赋值。

所以基于以上原理，在进行冗余逻辑优化和合并的基础上，可以增加最小面积优化方法，在资源估算时，以LUT4为base进行初步资源估算，在当前基础上，以LUT4之间的优化方法，我们可以以高维度LUT为Base的FPGA作为方向，**将多个LUT在合理的范围内进行合并**，合理的范围即在一个Module内的组合逻辑或连续赋值、阻塞赋值、非阻塞赋值等LUT合并至LUT6，从而实现以最小面积为目标的优化策略。

## 结果分析

### 快速资源估算模型无优化结果对比

### 快速资源估算模型增加优化结果对比

表3.1、表3.2分别为快速资源估算模型在5个基本Case下，未增加优化项与优化算法的估算结果与增加优化项及优化算法后的准确值**与估算时间的对比**，通过分析我们可以发现在不增加优化项及优化算法的情况下在使用资源量即电路逻辑复杂度较低的情况下准确度较好，但随着设计规模的增加电路复杂度的上升其预测资源准确度越来越差，由于对电路的过度精细化导致在面对设计规模较大的电路时由于内存原因导致无法得到估算结果，但是在时间性能上已经较综合布局得到较高的提升最小的设计时间提升约15倍，随着设计负责度的增加，在时间性能提升越明显，同时对比优化后结果可以得出首先由于设计规模较大无法计算的资源已经可以得到结果，且各个Case的估算结果准确度较好，LUT资源估算平均准确度达90.11%，FF资源估算平均准确度达88%，并且在时间性能上进一步提升，较未优化模型提升2倍，已基本满足在高密度FPGA原型验证系统划分阶段中对超大型设计进行资源估算的需求。

## 本章小结

**本章详细叙述了，在以第二章理论为基础的前提下**，使用Verific构建面向高密度FPGA原型验证系统RTLPartition的FPGA设计的快速资源估算模型，同时提供多种优化及约束方案，对模型进行约束及二次优化，使得对设计在FPGA上的资源估算精确度达90%以上，时间性能，同时为用户提供多种额外功能，如用户指定资源输入、对应不同FPGA的使用数量评估等作为商业软件对用户的友好性功能。同时也为后续基于人工智能的FPGA资源估算模型提供思路及开发基础。

## 特征提取

对于机器学习项目，决定预测准确度的上限并不取决于模型的算法的高深或模型算法的先进性，而决定模型预测上限的是特征的全面性、准确性和样本。所以特征的选择和提取尤为重要。

本章主要阐述基于第二章中提到特征提取方法，使用Verific的Parse-tree来对输入的RTL设计进行二次解析并提取所需要的信息作为后续机器学习的输入特征。

### Verilog HDL 语法特征分析

基于前一步骤实现的基于Verific的快速资源估算建模所获取的资源之间的映射关系，优化技巧等理论及实践基础，我们发现对于综合后技术映射的基本逻辑就是将Verilog HDL预计合理的使用FPGA上的器件来实现Verilog代码中所需要的数字逻辑，所以我们可以分析将Verilog HDL设计语言和器件间的典型关系，提取VerilogHDL中的语法特征，通过这些特征来构建机器学习模型，从而实现更加快速精确的大型FPGA设计所需要的FPGA片上资源数量，首先我们基于IEEE1801所给出的Verilog的技术文档，对VerilogHDL进行全面特征提取。

通过对Verilog HDL基本语法分析结合IEEE Verilog 标准、RTL样例设计，拟提取Verlog HDL语法特征对应关键字如表4.1所示。

### Verilog主要语法类型与关键字

#### 语法类型 语法关键字

端口 Input Output Inout

线网 Wire Reg

**逻辑运算符、算术运算符、关系运算符 + - \* / ^ | &**

连续赋值语句 Assign



Always过程块 Always

过程赋值语句 = < =

条件表达式 If else then

条件分支语句 case

循环语句 For While Repeat

函数语句 Function

特征提取方法

首先我们通过第二章的综合及技术映射原理已知Verilog HDL语句的综合原则及对应的FPGA片上器件的映射规则，所以我们基于以上原理，借助Verific的Parse-Tree对当前读入的Verilog设计进行二次解析，具体流程图如下：

特征提取器结构图

通过解析Verific读入设计时构建的Parse-tree对所有Verific进行提取主要包含以下小节所述类别。

端口信息

首先对于FPGA上的IO资源是通过RTL中Module的端口映射而来，所以我们首先提取Module的端口信息，对于一个Module的端口信息在Verific的Parse-Tree中存储在VeriModule类下的VeriDataDeclar类下面，所以我们通过重载Verific提供的Visitor虚函数来获取端口数据类型指针，通过解析VeriDataDeclar下的数据解析出端口数量包括Input、Output、Inout，同时通过解析VeriDataDeclar下的VeriDatatype数据类型，获取对应的端口位宽，并存储为InputWidth、OutputWidth、InoutWidth，从而完成对Module端口特征信息的提取。

线网信息

Verilog VHDL线网类型语法在RTL级描述中也是很重要的存在，基于第二章中描述的综合及技术映射理论可知，线网通常用于赋值语句左右，或被映射为寄存器结构，所以获取Verilog设计的线网类数据信息对于机器学习模型的训练而言应相对重要。

对于线网结构在Verilog HDL语法中主要体现为Wire和Reg，在Verific的Parse-Tree中线网结构被构造为VeriNetDecl类进行存储，我们通过重载Visitor基类来获取Module下的全部的VeriNetDecl类，通过解析可获得Reg和Wire的数量以及对应的位宽，分别存储为特征Reg、RegWidth、Wire、WireWidth，从而完成对Module线网信息的提取。

操作符及运算表达式

在Verilog HDL语法中，表达式是将操作符和操作数联合起来都是用的一种Verilog HDL语言结构，通过运算得到一个结果，从语义上说，这个结果值也可以看作操作数的函数值；而操作运算符的类型与C语言类似，如图4.3所示。

Verilog操作符图示

首先我们明确在逻辑运算符、算术运算符和关系运算符在进行逻辑综合后会被综合为相应的逻辑运算，所以符号表达式的特征信息也是关键特征，所以我们通过如下方式对表达式进行解析从而获得，首先整个表达式在Verific的Parse-tree中被存储在VeriExpression类下，通过解析VeriExpression类获得表达式存储类VeriOpenbinery类，通过判断VeriOpenbinery类下的Type类型获取运算符类型，如OR、AND、PLUS、AlterShift等，并存储至特征Map中，通过获取运算符左右两侧表达式，通过递归算法递归解析剩余运算符两次表达式，从而获取运算符两次表达式位宽，并存储至特征Map中，递归完毕后既可以完成对运算符表达式的特征提取，算法伪代码如图4.4所示。

运算符特征提取伪代码

连续赋值语句

连续赋值语句为Verilog HDL语法中比较重要的语句，常被综合为FF及LUT等关键器件，所以连续赋值语句的特征提取尤为重要，由于连续赋值语句常包含组合或嵌套式语法，如下图所示，通过IEEE标准中对Verilog连续赋值语句的定义可知，连续赋值语句左侧只能为线网结构即wire类型，而右侧可以包含逻辑运算符（&、|、~）、算术运算符（+、-、\*、/）、选择表达式（A?B:C）等多种语法嵌套组合。



由于需要解析嵌套语法且种类较多，所以采用递归算法对表达式进行解析，首先根据=号左右获取AssignStatement，由于左侧只能为线网结构，所以将表达式左侧存储为AssignRightWidth用以存储assign语句左侧位宽，右侧进一步根据符号进行拆分，若存在逻辑运算符则存储逻辑运算符数量如AND (&)、or (&) 等符号特征，并存储数量，则先以符号进行左右两侧拆分，若存在括号，则先进行括号匹配，对提取出的表达式进行进一步递归解析，同时计算符号两侧位宽，并保存为ANDRIGHTWIDTH、ANDLEFTWIDTH等；若表达式右侧为选择表达式，则存储选择表达式数量为QUESTIONCOLON特征，进一步解析该选择表达式分别提取if条件并存储为QUESTIONCOLONIF特征，若if条件存在逻辑或算术运算同理递归调用进行解析并存储至对应特征，else则存储至QUESTIONCOLONELSE特征，同时解析else的逻辑与算术运算，同时存储至对应特征中，then语句则存储至QUESTIONCOLONTHEN特征中，同时递归解析Then语句中的逻辑与算术运算并存储至对应的特征统计中，算法伪代码如图4.5所示。

连续赋值语句特征提取伪代码

过程结构体

Verilog HDL语法中每个过程块是由过程语句initial或always和其内容语句块组成的，由于initial语句是不可综合语句，所以在该特征提取方案中不考虑initial过程块。其中语句块主要由过程性赋值语句（包括过程赋值语句和过程连续赋值语句两种）和高级程序语句（包括条件分支语句和循环控制语句）这两种行为语句构成，其语法格式如下，如图4.6所示。

Verilog Always过程块语法格式

由于Always过程块主要受敏感事件列表控制，敏感事件列表是由一个或多个表达式构成，当存在多个表达式时用or将它们组合起来。

所以对于Always块的特征信息的提取，主要分为三部分：

获取Always语法块数量；

获取Always语法块敏感事件列表、并解析宽度；

获取Always语法块内容（由于语法块中包含语法种类较多，将在后续着重叙述）；

在Verific Parse-tree中always块构造存储在VeriAlwaysStatement类中，通过重载Visitor函数，来获取Parse-Tree中的Always结构体的数量，通过统计数量可以获得RTL设计中Always块的数量，通过解析获取到的VeriAlwaysStatement结构中的VeriEventExpress类获取Always的敏感事件列表，由于敏感事件列表由表达式构成，所以通过使用前述3）、4）所提及的递归方法来获取敏感事件表达式的宽度，并存储至特征AlwaysEventWidth中；从而完成对Always结构体数量与Always结构敏感事件列表的特征信息提取。

Always特征提取伪代码

过程赋值语句

Verilog过程赋值语句语法格式

过程赋值语句的语法格式如图4.7所示，其中赋值操作符可以是“=”或“=”，它们分别代表了阻塞型和非阻塞型赋值类型。过程赋值语句只能用于对变量型数据进行赋值操作，结合第二章的逻辑综合原理可知，赋值语句右侧的赋值表达式可以用逻辑表达，从而综合为逻辑资源，所以过程赋值语句的特征信息提取也相对重要，由于过程赋值语句中的赋值表达式可以使用任何合法的表达式，存在符号表达式等，所以我们在4.2.3、4.2.4、4.2.5的递归提取方法的基础上对过程赋值表达式进行特征提取，提取特征类别分别为，阻塞型赋值语句数量BlockAssign，阻塞型赋值语句左右位宽BlockAssignRightWidth、BlockAssignLeftWidth、阻塞型赋值语句左右线网数量BlockAssignRightNum、BlockAssignLeftNum；非阻塞型赋值语句NonBlockAssign、非阻塞型赋值语句左右位宽NonBlockAssignRightWidth、NonBlockAssignLeftWidth，其中符号表达式部分，将被存储在对应的符号表达式特征中提取过程与方法思路与连续赋值语句一致，可参见图4.5。由于过程赋值表达式主要被用于过程块中，所以4.2.5中对语法块内容的提取将在该部分体现。

If-else条件分支语句

条件分支语句对于硬件的行为描述及其重要，且常被综合为FF和LUT器件，是一种高级语言的抽象表达，故在特征信息提取时，也是重点关注的语法；

在VerilogHDL中的高级程序语句，常用语过程块中，图4.5中已经提及过程块中常由两种普通的过程赋值语句及高级程序语句，高级程序语句就包含分支语句和循环控制语句，分支语句包含条件分支语句及case条件控制语句，该小节主要分析if-else条件分支语句的特征提取，case条



件控制语句将在下小节进行分析。

通过Verilog HDL语法可知，if-else条件分支语句的作用是根据指定的判断条件是否满足来确定下一步要执行的操作。其主要语法形式如图4.8所示。

Verilog If条件语句语法格式

由于if-else条件分支语句常用语Always过程块中，所以我们在4.2.5模块的基础上对if-else条件分支语句进行解析，在Verific-Parse-tree中，条件分支语句被存储为VeriQuestionColon类中，通过解析该类可获得if-else条件语句块数量特征Questioncolon，，由于if-else条件语句存在多条件或多if-else条件嵌套组成，所以采用递归方式分别解析if条件、if条件内容、else分支内容、then条件内容，由于if条件内容为条件表达式，所以可以采用4.2.3模块进行解析，并存储特征QUESTIONCOLONIF；if条件内容为赋值表达式，所以可以使用4.2.6中对过程赋值语句的提取方法来提取，并存储至对应的过程赋值特征中，同理可对Else内容块及Then内容块解析，并提取特征QUESTIONCOLONELSE、QUESTIONCOLONTHEN。从而完成对if-else条件分支语句的特征提取。

If条件语句特征提取伪代码

Case条件分支语句

基于4.2.7中对Verilog HLD分支控制语句的分析可知，case分支控制语句与if-else语句都是用来实现分支控制选择，但是case分支控制语句是用来实现多路分支选择控制的分支语句，其在实现多路选择控制上更加简便和直观，其通常用于描述译码或状态机，其语法格式如图4.10所示。

Verilog Case语句语法格式

在Verific-Parse-tree中Case语句块被解析构造存储至VeriCase类中，通过统计该类数量可获得Case数量存储至CaseNum特征，VeriCase类中包括Case控制表达式及分支表到式及分支语句块，模块需要对该类及表达式进行解析，解析控制表达式可用4.2.4中所述方法进行递归提取，从而获得控制表达式数量CASECONDITIONNUM、控制表达式位宽CASECONDITIONWIDTH、以及分支数量CASTITEM，由于分支语句块内容多为赋值语句，所以可以使用4.2.5中模块进行对应特征提取。Case条件分支语句特征提取伪代码如图4.11所示。

Case条件分支语句特征提取伪代码

循环语句

与条件分支语句一样，循环控制语句也是一种高级Verilog HDL程序语句，Verilog提供的循环控制语句有forever、repeat、while、for四种循环语句，其中除去forever语句是一种无限的循环语句外，其余的三种循环语句都是根据指定循环条件或循环次数进行循环；循环控制语句语法格式如图4.12所示。

Verilog循环语句语法格式

对于循环语句其逻辑综合后为将循环体按循环次数展开，所以也需要对其提取特定特征，主要分为以下两部分即循环次数，循环体内容。

对于For循环体，由于其循环次数在for语句条件表达式部分，需要解析出for循环初始值，判断条件，补偿，从而计算出循环次数，for循环语句块在Verific-Parse-tree中被构造为VeriFor结构，其中循环次数部分需要二次解析，通过解析后通过计算初始值-终止值步长来获得循环次数，并存储于特征foritem，同时将for结构体数量存储于特征forblock，对于循环体而言，其内容可嵌套if-else、赋值表达式等多种语法，需要使用递归算法对循环体进行解析，使用4.2.4、4.2.6中所提及的算法进行特征抽取，同时需要对统计的数量再乘以循环体个数，并存储至特征内。

对于while循环体，由于其判决条件存在与其条件表达式中，所以需要通过二次解析条件表达式来获取终止值，同时需要通过判断终止值的变量表达式来计算循环体次数。

对于repeat循环体，由于repeat循环体的循环次数在其循环次数表达式中，只需对循环次数表达式进行二次解析获得循环次数即可，同理使用上述方法解析循环体特征信息，并存储至对应特征中；

对于forever循环语句，只需要将Forever语句中的forever循环体内容进行解析即可完成对forever特征的提取，通过以上方式完成对Verilog循环语句的特征提取。

循环语句特征提取伪代码





## 函数语句

Verilog HDL语言中也包含类似于其他编程语言中的函数概念一样的语法，函数（Function）的语法形式如下，如图4.15所示。

### Verilog HDL Function语法结构

函数定义结构只能出现在模块中，而不能出现在过程块内且函数必须至少有输入输出端口和双向端口。

在Verilog HDL语句中，函数代表了纯组合逻辑。函数在综合的时候，被展开成了内联（in-line）的代码。任何在函数中定义的局部变量都被当做纯粹的临时变量，仅被综合成导线。由于Function内部代表了纯组合逻辑，所以对于Function内部的逻辑表达及Function的端口及调用数量进行特征分析及提取；

在Verific的Parse-tree中Function被构造存储于VeriFunction类中，通过重载Visitor函数获取该RTL设计下的VeriFunction类，并统计其数量并存储至FunctionNum特征中，同时进一步获取Function的内部逻辑结构，同时对内部逻辑使用4.2.3、4.2.4、4.2.5模块中提到的方法对组合逻辑进行提取，并存储至对应的类中。从而完成对Function的特征提取。

### 特征器构建

#### RTL语法特征提取器流程图

通过4.2所述的特征提取方法构造对应的模块，使用C++设计并开发特征提取器程序，特征提取器框图如图4.16所示。通过C++开发测试，从而完成自动多Case特征提取，提取RTL的特征总数量为67。

涵盖VerilogHDL的所有关键语法及在逻辑综合和技术映射中重要的语法信息。包括逻辑运算如（or、shift）的位宽信息、算术运算符的位宽信息、连续赋值运算的位宽信息、阻塞及非阻塞赋值运算的位宽信息等。

### 本章小结

本章叙述了本文中面向机器学习进行资源估算中最为重要的一部分，本章通过结合第二章所述的特征提取方法，通过对Verilog HDL语言分析，结合其语法在逻辑综合、技术映射上的特征关系，完成特征提取器的设计，通过C++完成开发，实现对RTL级代码的特征提取，为下一步建立机器学习模型的建立打好基础。

### 构建机器学习模型及训练

本章将使用第五章构建的特征提取器对样本数据构建数据集，并对数据集样本进行分析，洞见特征数据间的数学联系及统计学关系，并尝试多种机器学习回归模型，选择表现较好模型进行进一步训练。

### 数据集分析

#### 数据集构建

由于机器学习本质上是使用大量的样本数据，通过计算机实现的数学模型，来探索并计算输入特征之间的数学关系，从而实现预测，所以需要大量的样本数据来构建数据集，所以我们使用300个Verilog HDL设计及其在Xilinx Vivado综合工具上综合后输出的资源利用率结果来作为基础数据集及预测标签。

VerilogHDL数据集来自GitHub、Opencore等开源网站，基本数据集涉及CPU、加法器、乘法器、GPU、机器学习等多种类型。部分数据集展示如图5.1所示，数据集主要描述数据如图5.2所示。

#### 部分基本数据集数据图示

#### 数据集基本概要

#### 数据集分析

#### 数据可视化

我们对数据集使用第四章构造的特征提取器进行特征提取后对数据进行可视化，将可以便于我们对数据进行进一步的分析，使用直方图来展





示数据，如图5.3所示。

特征数据集直方图

相关性分析

数据是事实或观察的结果，是对客观事物的逻辑归纳，是用于表示客观事物的未经加工的原始素材。数据相关性是指数据之间存在某种关系，如正相关，负相关。所以我们需要对数据进行相关性分析，指标为标准相关系数也称为皮尔逊系数 $r$ 。使用第四章所构建的特征提取器所提取的特征进行整体特征分析可以得到如图5.4所示的特征热力图与表5.1、表5.2、表5.3、表5.4所示的各个资源与RTL级特征间的皮尔逊相关系数。

特征热力图

RTL级特征与LUT资源的相关性数据

RTL级特征与FF资源的相关性数据

RTL级特征与BUFG资源的相关性数据

RTL级特征与IO资源的相关性数据

通过分析我们可以得到FPGA片上资源LUT对应的RTL级代码使用第四章所述的特征提取器所提取的特征之间的相关性关系，我们可以发现较多特征与LUT、FF、BUFG、IO资源具有典型相关性，进一步验证了通过第四章构造的特征提取器提取的特征基本符合第二章中逻辑综合原理与技术映射原理，证明该特征提取存在合理性，同时相关性分析也为我们下一步特征工程特征降维删除不相关特征提供参考及依据。

特征工程

本研究共计提取了67项特征，这些特征的数据取值范围差别很大，量纲也不同，如端口数量为个，而位宽大小为位且数量级不同等，没有办法进行比较。所以我们需要对特征进行进一步的预处理，使其能够更加适合在机器学习环境下进行预测。则需要构造特征工程流水线来帮助减少复用操作，从而在样本数据集发生更新时能够快速处理。

特征预处理

数据清理

大部分的机器学习算法无法在确实的特征上工作，所以我们要创建一些函数来辅助其完成，首先就是对于数据存在缺失的情况，为解决它我们选用三种解决办法：

放弃这些数据缺失的区域；

放弃整个属性；

将缺失值设置为某个值，比如0或者平均数、中位数等

通过对预测目标和特征属性的分析，由于该特征目标为预测准确设计所需FPGA资源数值，所以对于数据存在缺失的数据类型，我们选择放弃这些数据缺失的数据，从而不对预测结果产生影响。

处理文本和分类属性

由于特征中可能存在除数值属性外的其余数据表达形式的特征，由于机器学习模型是数学模型，所以我们需要将文本数据属性和具有分类含义的数据属性进行抽象处理：

使用OrdinalEncoder顺序编码对文本属性进行编码；

使用OneHotEncoder独热编码对分类属性进行编码；

特征缩放

最重要也是最需要应用到数据上的转换就是特征缩放。如果输入的数值属性具有非常大的比例差异，往往会导致机器学习算法的性能表现不



佳，当然也有极少数特例，比如提取特征中的端口数量和端口位宽就很有可能存在数量级上的差异。

在机器学习领域最常用的同比例缩放所有属性特征的两种常用方法：

1、最小-最大缩放也可称之为归一化，归一化的操作较为简单，即将样本数值重新处理后使其使其最终范围归于区间[0,1]之间，具体的实现方法是将该样本值减去样本数据集中的最小值比上样本最大值与样本最小值的差。

2、标准化方法，与归一化方法相比，标准化则完全不一样，其实现是首先减去平均值，所以标准化值得均值总是零，然后除以方差，从而使得结果的分布具备单位方差。与最小-最大缩放的区别是，标准化不将值绑定到特定范围，但是标准化的方法受异常值的影响更小。

转换流水线

由于上述特征预处理需要使用正确的顺序来执行，所以我们采用scikit-learn提供的pipeline类来支持我们实现上述对数据的转换，构造流水线转换器如图5.5所示。

特征预处理流水线

当数据集经过特征预处理流水线处理后，完成基本特征工程便具备了进入模型进行训练的基础。经特征工程处理后最终输入特征张量为61\*218，标签为使用Xilinx公司Vivado软件进行综合及布局布线后获得的准确资源利用率数值。

模型框架选择

模型训练开发流程

机器学习模型训练开发流程如图5.6所示，在完成数据分析，特征分析，特征工程后，我们需要先数据集进行划分选择合适的机器学习模型来进行预测，首先我们不知道选择哪种模型的时候，先对各种回归模型进行初步尝试，指标采用R2和RMSE进行评估，然后选择在初始参数及该特征下表现较好的模型框架进一步对模型参数进行配置。

初步模型选择

我们分别使用线性回归、决策树、SVM向量机、随机森林、深度神经网络来对数据集进行初步的训练，从而选择出两种有前途的特种进行进一步的训练和微调从而实现我们的预测目标。

首先我们对数据集进行划分，我们随机抽样20%的数据作为验证集，剩余80%作为训练集，为保证训练及验证的全面覆盖，我们在验证时使用5折交叉验证来对我们的模型进行预测验证。以下结果均采用标准参数对LUT资源进行尝试预测，预测结果如图5.5所示。

各模型尝试预测LUT资源结果

通过上图所示对多种机器学习模型进行原始参数初步尝试后，我们可以发现在该特征提取及样本集上线性回归、决策树的表现的初始模型预测表现较差，但是随机森林、MLP这两种模型在标准参数下存在较好的的预测性能，所以本文下面将着重以随机森林及深度神经网络来进行进一步的预测模型训练及参数调整。

模型训练

Keras 是一个具有高度模块化的神经网络库。基于 Python 语言，能以 TensorFlow 或Theano 等为后端运行。

Keras 能够非常快速的搭建网络模型，以便迅速的将使用者的想法进行实现。基于上述目的 Keras 具有以下特征，首先，其可以在 CPU 与 GPU 上无缝的切换运行；第二，拥有非常简洁且友好的调用接口，减小了研究人员的工作量；第三，模块化设计，使用者可以根据需求非常方便的配置自己的模型；第四，支持支持任意的网络架构。基于上述的优势，本文选择 Keras作为机器学习的基本框架。

机器学习模型训练及调优流程

随机森林

本节在5.3.1所尝试的随机森林模型的基础上对随机森林参数进行调优，并通过学习率曲线结合随机搜索与网格搜索对参数进行进一步优化，我们将主要调整影响度较大的n\_estimators与max\_depth参数用于寻找最优模型。模型训练调优流程如图5.6所示。



## LUT资源模型训练

### 随机森林预测LUT模型参数搜索学习曲线

### 随机森林LUT预测模型学习曲线

依据图5.6所示对随机森林预测LUT模型进行训练，首先通过多次迭代依据借由随机森林模型提供的特征贡献度对特征进行二次调优，通过放弃贡献度较低的模型来提升模型准确度，最终输入张量为59\*220。在完成特征二次调整的基础上对模型进行超参数调整，主要参数n\_estimators与max\_depth的学习曲线如图5.7所示，验证方法使用10折交叉验证，得到在n\_estimators取33与max\_depth取12时评价指标RMSE最优。

## FF资源模型训练

### 随机森林预测FF特征搜索学习曲线

### 随机森林FF预测模型学习曲线

依据图5.6所示对随机森林预测FF模型进行训练，训练流程与5.4.3.2方法一致，通过迭代训练与参数调优，主要参数n\_estimators与max\_depth的学习曲线如图5.9所示，验证方法使用10折交叉验证，得到在n\_estimators取89与max\_depth取7时评价指标RMSE最优。

## 人工神经网络

### 预测LUT资源模型训练

### 神经网络学习曲线，左图为超参数调优前学习曲线

依据图5.6所示流程图构建神经网络模型，为探究初始数据集在模型上的表现特征，通过对未进行超参数调整的模型进行1000训练可得如图5.11所示学习曲线，通过分析可发现初步训练模型存在过拟合情况，为解决过拟合情况，通过使用随机森林迭代构建的特征，并调整模型深度与宽度来解决该问题，分别使用随机搜索与网格搜索对模型隐藏层层数、神经元数、学习率及采样数进行调整，均采用5折交叉验证及提前终止回调用于终止搜索裁判，通过参数搜索可得神经网络模型如图5.12所示，神经网络包括输入及输出层、隐藏层共计6层，输入层宽度为59，由于为典型回归模型故输出层宽度为1，隐藏层共4层，对应神经元数分别为59、96、110、58，学习率使用搜索最佳学习率为0.000107，对模型进行200轮次训练。神经网络训练学习曲线如图5.13所示，通过观察学习曲线可以发现通过特征迭代与超参数调优已经解决过拟合问题，且拟合情况较好。

### 神经网络模型结构图

### 神经网络与LUT模型学习曲线

### 预测FF资源模型训练

依据图5.8所示流程图构建神经网络模型，使用神经网络模型预测FF资源的训练方法与5.4.3.1基本一致，分别通过随机搜索与网格搜索对神经网络模型进行空间搜索，得到如图5.14所示的神经网络结构，隐藏层共3层，对应神经元数分别为53、162、27，学习率使用搜索最佳学习率为0.00013，对模型进行235轮次训练。神经网络学习曲线如图5.15所示，**模型拟合度较好未出现明显的过拟合或欠拟合情况。**

### 神经网络模型结构图

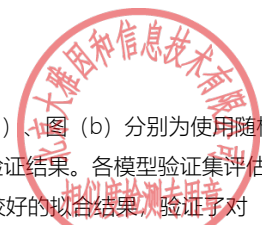
### 神经网络模型预测FF资源学习曲线

## 结果与分析

### 机器学习模型验证结果

### 模型评估

图5.16为机器学习模型在验证集上预测值与基准值的验证结果，X轴为验证集的准确值，Y轴为预测值。图（a）、图（b）分别为使用随机森林模型对LUT资源、FF资源预测的验证结果，图（c）、图（d）为人工神经网络模型对LUT资源、FF资源预测的验证结果。各模型验证集评估指标如表5.7所示，结合R方评估指标与图5.16我们可以发现机器学习模型在第四章构造的特征提取器基础上得到了较好的拟合结果，验证了对VerilogHDL语言特取的特征对LUT与FF资源的预测有较好的数学描述。但同时也能发现在LUT、FF资源的预测上人工神经网络这种对非线性特征



的学习能力较高的模型表现结果优于随机森林，对LUT预测和FF分别的模型确定性系数达73.4%、85.06%，证明模型已经可以较好的描述特征所代表的数学意义，同时均方根误差对LUT、FF资源分别为和。我们在测试数据集的基础上选取了模型评价指标较高的模型的结果进行分析，结果表明，在使用随机森林对RTL级设计RTL特征进行资源估算后与基准值进行对比，达到了31.3391、13.2000误差的精度，得到了一个较为满意的结果，同时由于该模型预测结果面向为高密度原型验证系统，主要应用于RTL-Partition阶段，基于其对FPGA片上资源利用率存在阈值，即常规情况下划分结果平均占用FPGA资源的50%~70%，且在划分系统程序中可以通过设定资源阈值控制来调整误差，所以当前模型输出的结果基本符合高密度FPGA原型验证系统的需求，同时除去精度外其更加聚焦于时间性能上的提升，通过统计RF、MLP的模型预测时间，我们可以明显看到机器学习在时间性能方面的巨大优势。

#### 本章小结

本章主要提出了使用机器学习来进行RTL设计在RTL级对其所需FPGA片上资源估算的方法，在对原始数据集进行特征缩放、特征标准化、特征降维等特征工程处理后分别使用RF、MLP进行回归预测，叙述了对多种机器学习模型在标准参数下对使用文本方法所构建的数据集的预测性能表现及对随机森林、深度神经网络进行进一步预测、参数微调等方法及结果的叙述，最终我们得到了对RTL级设计FPGA片上资源利用率预测符合预期且合理的模型。

#### 总结与展望

##### 工作总结

本文的主要研究方向集中在高密度FPGA原型验证系统软件设计CAD流程中的划分阶段的资源估算部分，研究主要分为三个方面：首先介绍了原型验证流程中FPGA的设计综合流程，包括逻辑综合原理，逻辑综合中Verilog HDL语法对应的逻辑综合原则，技术映射原理，现阶段工业界常用技术映射算法的描述，其次基于该原理结合高密度FPGA原型验证系统划分阶段需求设计并实现了快速资源估算模型，最后提出在EDA软件中结合机器学习方法，将资源估算与机器学习相结合，最终实现以下内容：

- 1.实现以传统思路框架为基础的快速资源估算模型，用户输入RTL级设计即可以快速得到设计所需要得FPGA片上资源数量如LUT、FF、BUFG等资源，有效提升了在超大型集成电路设计原型验证RTL划分阶段的迭代速度。
- 2.并在此基础上加入了合理的LUT优化算法，Bottom-To-Up算法，使该快速资源估算模型更加匹配实际划分需求及面向不同设计种类的优化方法及面向不同需求的优化方法，并增加可复用资源结构定义，并支持用户自定义功能user-define，使得原型验证划分阶段的迭代速度和实用性得到显著提升，有效提升约2.2倍，由于自主开发故拥有完善的C++输入输出接口，可集成在商业软件中，且实际应用至Player-Pro RTL Partition工业商用软件中，并在实际客户应用场景中得到检验。
- 3.以EDA与机器学习相结合的思路，结合逻辑综合及技术映射原理，构建面向RTL级集成电路行为描述的特征提取器，以Xilinx EDA设计软件为标签，训练了随机森林、SVM支持向量机、人工神经网络等多种机器学习模型，使运行速率提升15倍，平均准确度达75%，同时验证了EDA软件与人工智能相结合的可行性，进一步提升了在超大规模集成电路设计FPGA原型验证的划分阶段的迭代速度，为EDA与原型验证流程相结合的研究打下基础。

##### 工作展望

面向RTL级设计的FPGA快速资源估算问题结合了图论、可行性问题、逻辑分解、逻辑优化等众多方面特性，具有一定挑战性，随着设计规模的逐渐增大，在验证方面的需求特性也在增加，如何能有效的快速且准确的得到RTL设计所需的FPGA片上资源，并能有效合理的支持原型验证系统的RTL-Partition划分是未来EDA流程中将面临的一大问题。

本文虽在这方面做了一些研究和实现，但仍需在一下方面做出进一步的改进：

- 1.目前虽然实现快速资源估算模型，并增加了符合实际映射原理的优化及约束选项，但是在特殊的FPGA板上器件映射计算方面依然存在较大误差，如BRAM、LUTRAM等先进器件，所以可以考虑后续在这方面增加进一步的研究，使得资源估算更加精确。
- 2.虽然在本文中验证了机器学习方法与资源估算相结合的可行性，但是还处于初步阶段，精确度及估算器件类型上还有待增加，如可参照文章【】中提到的多级估算方法，将初步得到的一般器件估算结果进行二级抽象，得到更高级别的资源类型。Comment by Administrator: 参考文献
- 3.在面向资源估算的机器学习模型中增加更高层级抽象特征，增加实际模型泛化率，可参照文章【】利用HLS高层次综合方法快速构建样本设计从而增加模型训练样本数量，使得机器学习模型的泛化程度更高更有效。Comment by Administrator: 参考文献

4.神经网络资源估算模型还可以进一步增加不同设计在不同约束下进行综合的样本,使得机器学习模型可以支撑不同约束下的资源估算结果,如面积优化、功率优化等约束选项。

#### 参考文献

- Marcin Kubica , Adam Milik , Dariusz Kania. Technology mapping of multi-output function into LUT-based FPGA[J]. IFAC PapersOnLine, 2018, 51(6) : 107-112.
- Heyse K , Farisi B A , Bruneel K , et al. TCONMAP: Technology Mapping for Parameterised FPGA Configurations[J]. Acm Transactions on Design Automation of Electronic Systems, 2015, 20(4):1-27.
- Huang G , Hu J , He Y , et al. Machine Learning for Electronic Design Automation: A Survey[J]. 2021.
- 吴滔, 林铠鹏. 高密原型验证系统解决方案 (上篇) [J]. 中国集成电路, 2021, 30(09):48-55+69.
- 吴滔, 林铠鹏. 高密原型验证系统解决方案 (上篇) [J]. 中国集成电路, 2021, 30(10):27-33.
- 张彪. 一种基于多FPGA的逻辑划分方法的研究与实现[D]. 西安电子科技大学, 2018.
- Hung W N , Sun R . Challenges in Large FPGA-based Logic Emulation Systems. 2018.
- Li B , Qi Z , Tang Z , et al. High quality hypergraph partitioning for logic emulation. 2021.
- Voss N , Quintana P , Mencer O , et al. Memory Mapping for Multi-die FPGAs[C]. 2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM). IEEE, 2019.
- 耿荣. EDA 虚拟仿真实验系统的设计与实现[D]. 内蒙古大学, 2019.
- 李玉梅. FPGA工艺映射算法的研究与分析[D]. 北京交通大学, 2010.
- Dai S , Zhou Y , Zhang H , et al. Fast and Accurate Estimation of Quality of Results in High-Level Synthesis with Machine Learning[C]. 2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM). IEEE, 2018.
- Makni M , Baklouti M , Niar S , et al. Hardware Resource Estimation for Heterogeneous FPGA-based SoCs[C]. The 32nd ACM Symposium on Applied Computing (SAC). ACM, 2017.
- Meeuws R , Galuzzi C , Bertels K . High Level Quantitative Hardware Prediction Modeling using Statistical methods[C]. International Conference on Systems, Architectures, MOdeling and Simulation (SAMOS XI). IEEE, 2011.
- Kubica M , Opara A , Kania D . Technology Mapping for LUT-Based FPGA[M]. Lecture Notes in Electrical Engineering, 2021.
- 刘福奇, 刘波. Verilog HDL应用程序设计实例精讲[M]. 1. 电子工业出版社, 2009.
- Makrani H M , Farahmand F , Sayadi H , et al. Pyramid: Machine Learning Framework to Estimate the Optimal Timing and Resource Usage of a High-Level Synthesis Design[J]. IEEE, 2019.
- Schumacher P , Jha P . Fast and accurate resource estimation of RTL-based designs targeting FPGAs[C]. Field Programmable Logic and Applications, 2008. FPL 2008. International Conference on. IEEE, 2008.
- Que Y , Kapre N , Ng H , et al. Improving Classification Accuracy of a Machine Learning Approach for FPGA Timing Closure[C]. IEEE International Symposium on Field-programmable Custom Computing Machines. IEEE, 2016.
- 洪耀球. 船舶油耗预测的机器学习算法比较研究[J]. 济南大学学报(自然科学版), 2021, 35(06):546-551. DOI:10.13349/j.cnki.jdxbn.20210513.005.
- 孟子流, 李腾龙. 机器学习技术发展的综述与展望[J]. 集成电路应用, 2020, 37(10): 56-57.
- 洪耀球. 船舶油耗预测的机器学习算法比较研究[J]. 济南大学学报(自然科学版), 2021, 35(06):546-551. DOI:10.13349/j.cnki.jdxbn.20210513.005.





戴源,白雨鑫,张伟,陈鑫.基于Vivado HLS的硬件设计效能评估[J].电脑知识与技术,2021,17(19):1-4.DOI:10.14004j.cnki.ckt.2021.1845.

郑玉杰,梁鑫斌,张起,孙文博,施童超,杜鹃,孙宽.基于分子指纹及机器学习回归模型的有机光伏材料效率预测[J].材料导报,2021,35(08):8207-8212.

刘斌,何进荣,耿耀君,王最.并行机器学习算法基础体系前沿进展综述[J].计算机工程与应用,2017,53(11):31-38+89.

雷国平,肖科,罗秀英,杨森,姚佳佳.基于机器学习的基础算法研究综述[J].卫星电视与宽带多媒体,2020(08):18-19.

路宝珠,杨海钢,郝亚男,张茉莉,崔秀海.基于区域重组的异构FPGA工艺映射算法[J].计算机辅助设计与图形学学报,2012,24(08):1027-1037.

王沐贤,丁小欧,王宏志,李建中.基于相关性的多维时序数据异常溯源方法[J].计算机科学与探索,2021,15(11):2142-2150.

王进,孙万彤.基于相关性分析的多标签特征选择方法[J].重庆邮电大学学报(自然科学版),2021,33(06):1024-1037.

周志华. 机器学习[M]. 清华大学出版社, 2016.

P. Bjureus, M. Millberg and A. Jantsch, "FPGA resource and timing estimation from Matlab execution traces," Proceedings of the Tenth International Symposium on HardwareSoftware Codesign. CODES 2002 (IEEE Cat. No.02TH8627), 2002, pp. 31-36, doi: 10.1145/774789.774797.

[1] Brandolese C , Fornaciari W , Salice F . An area estimation methodology for FPGA based designs at systemc-level[C].Design Automation Conference, 2004. Proceedings. 41st. IEEE, 2004.

Pilato C , Cattaneo R , Durelli G , et al. A2B: a Framework for the Fast Prototyping of Reconfigurable Systems.

Hara Y , Tomiyama H , Honda S , et al. CHStone: A benchmark program suite for practical C-based high-level synthesis[C].Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on. IEEE, 2008.

Pan S J , Qiang Y . A Survey on Transfer Learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(10):1345-1359.

2021. Vivado ML Standard Edition. <https://www.xilinx.com>.

Pedregosa F , Varoquaux G , Gramfort A , et al. Scikit-learn: Machine Learning in Python[J]. 2012.

Reagen B , Adolf R , Shao Y S , et al. MachSuite: Benchmarks for accelerator design and customized architectures[C] IEEE International Symposium on Workload Characterization. IEEE, 2014:110-119.

Ouaiss I , Vemuri R . Hierarchical memory mapping during synthesis in FPGA-based reconfigurable computers[C]. Design, Automation & Test in Europe, Conference & Exhibition. IEEE, 2001.

Daneshi M , Fard R T , Zomorodian Z S , et al. Development of a hybrid machine-learning and optimization tool for performance-based solar shading design[J]. 2022.

Martin S . Estimation of Conditional Random Coefficient Models using Machine Learning Techniques[J]. Papers, 2022.

上海国微思尔芯技术股份有限公司. 一种逻辑资源评估方法:CN20201119075.3[P]. 2021-01-15.

普通图书示例:

广西壮族自治区林业厅. 广西自然保护区[M]. 北京: 中国林业出版社, 1993.

蒋有绪, 郭泉水, 马娟等. 中国森林群落分类及其群落学特征[M]. 北京: 科学出版社, 1998.

唐绪军. 报业经济与报业经营[M]. 北京: 新华出版社, 1999: 117-121.

赵凯华, 罗蔚茵. 新概念物理教程: 力学[M]. 北京: 高等教育出版社, 1995.





汪昂. (增补)本草备要[M]. 石印本. 上海: 同文书局, 1912 .

CRAWFORD W, GORMAN M. Future libraries: dreams, madness, & reality[M]. Chicago: American Library Association, 1995.

International Federation of Library Association and Institutions. Names of persons: national usages for entry in catalogues[M]. 3rd ed. London: IFLA International Office for UBC, 1977.

O'BRIEN J A. Introduction to information systems[M]. 7th ed. Burr Ridge, Ill.: Irwin, 1994.

ROOD H J. Logic and structured design for computer programmers[M]. 3rd ed. Watertown: BrooksCole Thomson Learning, 2001.

论文集、会议录示例:

中国力学学会. 第3届全国实验流体力学学术会议论文集[C]. 天津: 南开大学出版社, 1990.

ROSENTHALL E M. Proceedings of the Fifth Canadian Mathematical Congress, University of Montreal, 1961[C]. Toronto: University of Toronto Press, 1963.

GANZHA V G, MAYR E W, VOROZHTSOV E V. Computer algebra in scientific computing: CASC 2000: proceedings of the Third Workshop on Computer Algebra in Scientific Computing, Samarkand, October 5-9, 2000[C]. Berlin: Springer, c2000.

科技报告示例:

U. S. Department of Transportation Federal Highway Administration. Guidelines for handling excavated acid-producing materials, PB 91-194001[R]. Springfield:U. S.Department of Commerce National Information Service, 1990.

World Health Organization. Factors regulating the immune response: report of WHO Scientific Group[R]. Geneva: WHO, 1970.

学位论文示例:

张志祥. 间断动力系统的随机扰动及其在守恒律方程中的应用[D]. 北京: 北京大学数学学院, 1998.

CALMS R B. Infrared spectroscopic studies on solid oxygen[D]. Berkeley: Univ. of California. 1965.

专利文献示例:

刘加林. 多功能一次性压舌板: 中国, 92214985. 2[P]. 1993-04-14.

河北绿洲生态环境科技有限公司. 一种荒漠化地区生态植被综合培育种植方法: 中国, 01129210.5[POL]. 2001-10-24[2002-05-28].  
<http://211.152.9.47/sipoasplzljshyjs-yx-new.asp?recid=01129210.5&leixin>.

KOSEKI A, MOMOSE H, KAWAHITO M, et al. Compiler: US, 828402[POL]. 2002-05-25 [2002-05-28].  
<http://FF&p=1&u=metahtmlPTOsearch-bool.html&r=5&f=G&1=50&co1=AND&d=PGOI&sl=IBM.AS.&OS=ANIBM&RS=ANIBM>.

专著中析出的文献示例:

国家标准局信息分类编码研究所. GBT 2659-1986世界各国和地区名称代码[S]全国文献工作标准化技术委员会. 文献工作国家标准汇编: 3.北京: 中国标准出版社, 1988:59-92.

韩吉人. 论职工教育的特点[G]中国职工教育研究会. 职工教育研究论文集. 北京: 人民教育出版社, 1985:90-99.

BUSECK P R, NORD G L, Jr., VEBLEN D R. Subsolidus phenomena in pyroxenes[M] PREWITT C T. Pyroxense. Washington, D.C.: Mineralogical Society of America, c1980: 117-211.

FOURNEY M E. Advances in holographic photoelasticity [C] American Society of Mechanical Engineers. Applied Mechanics Division. Symposium on Applications of Holography in Mechanics, August 23-25, 1971, University of Southern California, Los Angeles, California. New York: ASME, c1971 : 17-38.



MARTIN G.. Control of electronic resources in Australia[M]PATTLE L W, COX BJ. Electronic resources: selection and bibliographic control. New York: The Haworth Press, 1996: 85-96.

期刊中析出的文献示例:

李炳穆. 理想的图书馆员和信息专家的素质与形象[J]. 图书情报工作, 2000(2):5-8.

陶仁骥. 密码学与数学[J]. 自然杂志, 1984,7(7):527.

习亚洲地质图编目组. 亚洲地层与地质历史概述[J]. 地质学报, 1978, 3: 194-208.

DES MARAIS D J, STRAUSS H, SUMMONS R E, et al. Carbon isotope evidence for the stepwise oxidation of the Proterozoic environment [J]. Nature, 1992, 359: 605-609.

HEWITT J A. Technical services in 1983[J]. Library Resource Services, 1984, 28(3): 205-218.

报纸中析出的文献示例:

丁文祥. 数字革命与竞争国际化[N]. 中国青年报, 2000-11-20(15).

张田勤. 罪犯DNA库与生命伦理学计划[N]. 大众科技报, 2000-11-12(7).

电子文献(包括专著或连续出版物中析出的电子文献)示例:

江向东. 互联网环境下的信息处理与图书管理系统解决方案[JOL]. 情报学报, 1999, 18(2): 4[2000-01-18].

萧钮. 出版业信息化迈入快车道[EBOL]. (2001-12-19) [2002-04-15]

CHRISTINE M. Plant physiology: plant biology in the Genome Era[JOL]. Science, 1998, 281:331-332[1998-09-23].

METCALF S W. The Tort Hall air emission study[COL]The International Congress on Hazardous Waste, Atlanta Marriott Marquis Hotel, Atlanta,Georgia, June 5-8, 1995: impact on human and ecological health[1998-09-22].

TURCOTTE D L. Fractals and chaos in geology and geophysics[MOL]. New York: Cambridge University Press, 1992[1998-09-23].

Scitor Corporation. Project scheduler[CPDK]. Sunnyvale, Calif. : Scitor Corporation, c1983.

致谢

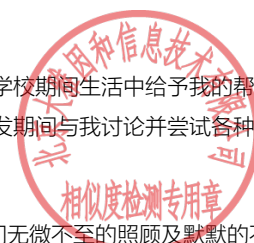
时光如梭, 光阴如剑, 转眼间三年的研究生生活即将结束, 也同时意味着十九年的求学生涯将暂时告一段落。在此论文完成之际, 衷心的感谢我的导师赖睿教授、微电子学院EDA研究院院长游海龙教授, 感谢二位教授将我带入了微电子EDA领域, 二位老师知识渊博, 治学严谨, 科研方法先进, 从最初的选题, 论文中期时研究方法的创新, 到最后论文完成时的耐下心来知道, 均得益于教授的谆谆教导和悉心帮助。

感谢祁仲冬老师对我编程能力的指导, 在程序遇到各种疑难问题时, 祁老师总是非常耐心的为我调试和解决。在论文的编写过程中, 感谢余丽艳老师在百忙之中抽出时间与我讨论, 提出许多宝贵的建议。

还要感谢实习企业深圳国微集团及上海国微思尔芯股份有限公司, 能够给我机会让我更深入的接触原型验证系统及EDA软件, 感谢在外实习期间的指导与认可。同时也很感谢团队中的每一个成员, 感谢同样毕业于西电这个大家庭的孙小霞、张云飞、陈辰、周晨等几位组内同事, 在研发与工作中给予我极大的帮助与指导, 还有公司的指导老师王宁、刘瑞华, 在研发与工作中给予我莫大的帮助, 还有惠茹、Cindy、KP、Zyee、Amir、Leew、Win等同事在实习期间给与的帮助与指导。

同时也要感谢同窗三年的室友们, 认认真真的王熙铭, 聪明机智的张江涵, 爽朗大气的赵子凡, 感谢他们在校期间生活中给予我的帮助与支持, 并与我一起在求学的道路上共同成长。朝夕相处定不相忘。也感谢学校实验室的李本正博士, 总是在我研发期间与我讨论并尝试各种解决方案。

最后, 谨以此文献给我的父母, 他们赋予了我宝贵的生命, 含辛茹苦的将我养育成人。求学二十载, 感谢父母们无微不至的照顾及默默的不求



回报的付出，虽然他们年岁渐高,身体也大不如从前,但是他们对我的爱却没有丝毫减少。无论何时何地,他们都是我最坚实的后盾和最温暖的港湾。

#### 作者简介

#### 基本情况

张曦，男，陕西西安人，1994年7月出生，西安电子科技大学微电子学院软件工程专业2019级硕士研究生。

#### 教育背景

20013.09 ~ 2017.07杭州电子科技大学，本科，专业：信息对抗技术

2008.08 ~ 西安电子科技大学，硕士研究生，专业：软件工程

#### 攻读硕士学位期间的研究成果

#### 发表学术论文

#### 申请（授权）专利

樊沁春、张曦、李楠. 专利名称: 基于深度学习的布局后布线违例预测方法及可读存储介质:中国,专利号202011437109.3 [P]. 2020.12.11.

张曦. 专利名称:一种将超图结构转RTL级HDL文件的方法及装置: 中国, 202110360258.2 [P]. 2021.04.02.

郭广鑫、张曦、陈灿. 专利名称: 一种检查Verilog时序模块驱动冲突的方法:中国,利号202011437109.3 [P]. 2020.12.11.

#### 参与科研项目及获奖

XXX项目, 项目名称, 起止时间, 完成情况, 作者贡献。

## 说明：

- 1.文献相似度 = 送检文章中与检测范围所有文献的相似字符数/送检文章正文字符数
- 2.去除参考文献相似度 = 送检文章中检测范围所有文献（不包括参考文献）的相似字符数/送检文章正文字符数
- 3.去除本人已发表论文相似度 = 送检文章中与检测范围所有文献（不包括自引）的相似字符数/送检论文正文字符数
- 4.单篇最大相似度：送检文章与某一文献的相似度高于全部其他文献
- 5.正文字符数：送检文章正文部分的总字符数，包括汉字、非中文字符、标点符号、阿拉伯数字（不计入空格），正文不包括关键词、目录、图片、附录、参考文献等

