Machine Learning Based Framework for Fast Resource Estimation of RTL Designs Targeting FPGAs

Abstract

Introduction

1.FpGA-based 设计、验证、需要对设计在FPGA占用资源进行评估；

FPGA-targeted designs

Inaccurate HLS estimates prevent designers from performing meaningful design

space exploration without resorting to the time-consuming downstream implementation process FPGA

FPGA-based emulator

在HLS论文introduction基础上，增加EMU、原型验证关于对FGPA资源利用估计的问题需求场景。

发展趋势。FPGA结构简介。RTL设计用FPGA实现的过程简介。本工作的意义（快速估计帮助设计 人员，可用于RTL划分指导）。本工作主要贡献。 Related Works 列举一些RTL资源估计、HLS资源估计论文 说明随着综合技术和FPGA的发展，早期的方法已经不能取得良好的结果。 大多数工业设计的HDL语言并非HLS得到（待考证）。

2.Backgroud与思路

2.1 问题

2.2 related work

2.3 Framework and Flow 思路

本文主要工作流程与机器学习框架结构

3. Data Processing And Analysis

Data Processing And Analysis

数据收集 数据来自opencores、github、公开的工业设计、HLS生成等。覆盖图像处理、神经网络、DSP、编码解 码器、CPU核等多种应用。数据规模也有大有小，给出每种资源最大最小平均值表格。 特征提取 使用Verific综合，分别在static elaborate后与elaborate后的这两级进行特征提取。（介绍Verific，阐述 这两种综合的作用）。提取了哪些特征，进行了什么样的处理（分为特征类别、简述、特征个数列表 格）

4.Estimation Models 随机森林

简要介绍原理及应用方式 神经网络 简要介绍原理、与随机森林的对比，及应用方式 多级学习 分为三级： 对DSP和BRAM的预测 加入上一级结果后对LUT与FF的预测 对LUT与FF预测结果的修正（提取出一些重要特征后，采取人工设定参数或者自动拟合的方式） 多级结果更加精确和分析其原因。

5.Experiments

在几种目标平台（Virtex UltraScale，Virtex UltraScale等）上进行测试。 分别采用上面提到的几种模型，对比效果。 精度衡量采用一篇论文中采用的relative absolute error (RAE)，介绍其计算方式。 采用5叠交叉验证的方式来测试，每次随机一部分作为测试数据，最后取精度平均值。 分别对LUT、FF、BRAM、DSP进行预测。 加速效果对比计算所有测试集case平均耗时，估计是百倍到千倍的提升。 给出训练用时。 目前交叉验证rae结果为 LUT：51% FF： 28% 后面尽力将其降低到20%以

6. results and Applications

Our proposed \*\*\*\*\*\* approach has several applications; here, we present two important ones.

增加一个针对RTL-parttition，应用的场景讨论。

**摘要**

使用FPGA对电路进行硬件仿真已成为芯片设计的重要环节。但随着设计的规模剧增，如何将超大型集成电路设计划分至多FPGA系统中成为了一个关键挑战。在电路设计流程早期快速估算每个子电路模块所需要的FPGA片上资源数量将为合理划分电路提供重要依据，同时也可以帮助设计人员调整电路。本文提出了一种使用机器学习方法快速预测RTL设计消耗的片上资源的框架。我们广泛收集了RTL设计作为数据集，分别从Verific工具解析后的源代码和符号门两级中提取特征并分析其作用，针对性的训练了一种三阶段的ANN模型进行预测。最终在具有合理的精度的同时取得了xxx倍的加速效果。

**1介绍**INTRODUCTION

**2动机**MOTIVATION

论文Challenges in Large FPGA-based Logic Emulation Systems中提到当前的FPGA系统电路划分中包含两种主要的flow，根据techmap和partition的先后顺序分为网表级划分与RTL划分，如图所示

图

RTL划分具有许多的优势，包括可以并行综合、含有电路边界信息指导综合、包含一些高层级信息指导划分。但是其关键问题是没有进行综合因此缺少资源使用量信息来指导划分。因此我们在研究时提出本文的解决方案以帮助在RTL级快速的获取大致的资源使用量信息，使得RTL划分可以顺利进行下去。这种应用场景下并不需要预测器具有非常高的精度，因为电路划分往往会留出30%~50%的资源裕量以避免布线拥塞等实现失败状况，同时划分前由RTL文件转化生产的超图往往有上万个节点，资源偏差较大的小部分节点往往只有微乎其微的影响。

同时RTL阶段快速的资源估计有助于在早期对设计进行评估，指导工程师修改设计。

**3数据处理**DATA PROCESSING AND ANALYSIS

为了使用机器学习完成RTL级资源预测，我们构建了一个数据集，它包含了288个Verilog语言的RTL设计。为了保证预测器的鲁棒性和训练质量，我们的数据集里的设计覆盖面非常的广泛。它们来自著名的FPGA开源IP核网站opencores、开源社区github中的RTL项目、公开的工业设计。我们也引入了著名的HLS基准CHStone 与Machsuite，并利用 Vitis\_HLS工具进行高级综合生成了对应的RTL设计。这些设计覆盖图像处理、神经网络、DSP、编码解 码器、CPU核等多种应用。各不相同的数据来源、应用类型、设计规模使得不同的设计间HDL代码特点风格差异巨大，这给训练与预测带来了很大的挑战。表中显示了数据集中设计综合后的实际资源使用量情况的一些整体特征。

进一步的，我们对于这些设计提取了可能的特征来使机器学习可行。 我们使用Verific® parser/elaborator tool对设计进行解析。

（--------------------------对verific起到的功能的介绍-------）

我们使用Vivado来对设计进行实现，以得到它们的资源使用量情况，即训练数据的标签。Vivado的综合工具有多种可自定义的配置参数，以及一些已经预先设定好的综合模式，例如最小化面积，或最快运行时间。在本文中，我们选择使用默认模式来进行操作，如果想要在其他模式下使用，可以使用类似的方法重新训练模型。尽管真正的资源使用量需要在工艺映射之后才可以获得，但是在综合之后报告的资源使用情况与其只有极小的差距，所以为了贴近真实的设计情况和节约时间，我们采用综合后Vivado报告的资源使用量作为标签。

首先我们利用集成电路设计相关领域的知识，提取了与我们的预测任务相关可能相关的大量特征。之后我们计算每一对特征的泊松相关系数，对于相关性较大的一组特征我们只保留一个。我们也计算了每一个特征与每一种目标资源的相关系数，删除了与任何资源相关性都很小的特征。最终，特征提取产生了53个较高相关性的特征值来反映资源的实际使用情况。其中与每一种类的资源相关性较高的几个特征描述如下

LUT：

FF：

DSP：

BRAM：memory深度、位宽度、写/读端口数以及端口的异步/同步性质。因为一般大规模的，只有一个写端口的同步电路中的reg才会被综合成为BRAM。而其他情况容易被综合成为LUTRAM，以提高性能。

**4预测模型**ESTIMATION MODELS

我们使用回归模型来预测LUT、FF、BRAM、和DSP的使用量情况。回归是一种著名的统计监督学习技术，它给出特征向量和标签间的关系。在本文中我们模型的输入是一个给定的向量x=[]，而输出是一个4维的向量y=[]。我们可以采用一次性输出y向量的多任务预测，也可以使用4次单任务预测分别得到y的4个维度值。[]中提到，单任务模式在FPGA资源预测中结果更好，本文也采用单任务的回归模型。有各种各样的机器学习模型可以完成这一项任务。在本文中我们分别使用随机森林和人工神经网络以及一种多阶段神经网络对四种资源进行预测。

1. 随机森林

随机森林回归（RFR）[15]是一个元估计器。它在一组数据集的子样本上拟合出一个回归决策树的值。增加森林中的树的数量和深度可以提高回归的精度。由于最佳分割的决策点的数量增加，因此 ML模型的大小变得更加广泛，需要更多的调整时间。

1. 线性回归
2. ANN

人工神经网络（ANN）由一个输入层、一系列隐藏层和一个输出层组成。每个隐藏层包含一组节点，这些节点在层中以加权连接排列。每一层使用前一层的输出作为输入。激活函数为模型添加了非线性性。根据文章【】【】中提到的，因为我们模型的输出范围较宽（尤其是lut与ff），以及实验结果的证明，使用ReLU作为激活函数，而不是Sigmoid或者Tanh会取得更好的结果。

针对于我们的预测问题，因为特征数量相对于大规模的神经网络来说较少，而且训练集的数据样本量也不多，在我们的实验过程中发现，如果使用深层的网络结构则很容易产生过拟合的问题。因此我们的人工神经网络结构仅有几个全连接的隐藏层。

与线性回归与随机森林不同，ANN可以表示更复杂的非线性结构，同时也需要更大样本量和更长时间的训练，但是可以取得更好的预测结果。

1. 多阶段ANN

为了进一步的提高预测精度，我们采用了一种三阶段的模型来完成预测任务，流程如图所示。

图

由于DSP与BRAM资源的使用情况会极大的影响LUT与FF的数目。例如与不使用任何DSP的综合方式相比，使用一个DSP后会减少几百的LUT数量。并且决定DSP与BRAM的相关性特征数量较少，预测的波动和误差相对较小。因此我们在第一阶段中先使用ANN预测DSP与BRAM使用量。在第二阶段开始前，将预测的DSP与BRAM结果作为两个新的特征维度加入到输入向量x中，注意此时训练集中的数据尽管已知实际的资源使用量，但是加入x向量的仍然是预测值，这样保持训练与测试集的一致可以得到更好的模型性能，实际运行的效果也证明了这一点。之后再在新的输入向量基础上训练两个ANN分别预测LUT与FF的使用量。第三阶段中会根据对于预测结果的一些分析以及集成电路相关知识对结果进行修正。具体如下：

BRAM：通常为了提高电路性能，小规模的memory和reg使用不会被综合成BRAM，而ANN预测结果中使用了小规模memory的设计BRAM预测结果偏大，大规模的设计预测结果偏小，因此根据memory和reg的规模对结果进行修正。

DSP：与BRAM类似，我们将运算规模大设计的结果增大，并缩小了小规模运算的设计的结果。

LUT：通常来说，电路中 LUT数目不会超过逻辑门与一位数据选择器的数量总和。即使存在触发器时。因此我们将预测值超过这个和的减小为这个值。

FF：在没有使用DSP与BRAM的情况下，在Verific总和后得到的vdffrs值往往和实际的FF使用量比较接近，而且在设计规模较大时越接近。因此我们对ANN结果和vdffrs值做加权平均，设计规模越大，vdffrs权重越高。

与一般的ANN相比，这种网络结构可以得到更精确的结果。

**5实验结果**EXPERIMENTS

我们采用第x节中描述的模型，在Python环境中利用keras【】和scikit-learn【】库完成实验。所用的设计均使用Vivado2021.2目标vu440xxxx，采用默认的综合选项完成综合。实验在平台Intelxxxx，xxxGHz上完成。

对于模型质量的评估，因为实际值中含有0值，因此RMSE，XXX等相对误差无法使用，我们选取采用RAE（相对绝对误差）作为指标，公式如下：（解释公式中变量）

对全部的训练数据进行5折交叉测试：将全部数据随机分成大小相等的5份，每次选择一份数据作为测试集，用剩下数据进行训练，在ANN迭代一定的次数时终止训练。最终取每次在测试集上预测结果RAE值的平均值评价整个模型。将线性回归、随机森林、ANN、前两阶段ANN、完整三阶段ANN的RAE的平均值及标准差展示在下表中：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | LUT | FF | DSP | BRAM |
| LG |  |  |  |  |
| RF |  |  |  |  |
| ANN |  |  |  |  |
| 2ANN |  |  |  |  |
| 3ANN | 0.33 (+/- 0.14) | 0.38 (+/- 0.12) | 0.42 (+/- 0.20) | 0.24 (+/- 0.09) |

可以看到，相比于一般的神经网络模型，分两阶段预测四种资源可以取得更好的结果，而对结果的修正也能起到一定的作用。

所有样本的实际值与预测值散点图如下所示，可以看出我们的模型对于XX的数据结果并不是很好，可能的原因是XX。实际上，XXX情况经常导致预测偏差较大。

流程耗时加速倍数随LUT用量散点图如下

图

一个case使用Vivado综合耗时在1~10min。而本流程中最大case耗时也不会超过2s。可以看到本文中的资源使用量预测流程，加速效果非常明显，这种速度下允许设计人员随时查看大致的资源使用量，以及指导划分器快速评估当前划分方案。