文本复制检测报告单(全文标明引文)

№:ADBD2020R_20200427203645450760087749 检测时间：2020-04-27 20:36:45

检测文献： 基于进化算法的FPGA脉动阵列快速硬核布局方法

作者： 张年崧

检测范围： 中国学术期刊网络出版总库
中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库
中国重要会议论文全文数据库
中国重要报纸全文数据库
中国专利全文数据库
图书资源
优先出版文献库
大学生论文联合比对库
互联网资源(包含贴吧等论坛资源)
英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)
港澳台学术文献库
互联网文档资源
源代码库
CNKI大成编客-原创作品库
个人比对库

时间范围： 1900-01-01至2020-04-27

检测结果

去除本人已发表文献复制比： 1.7% **跨语言检测结果：** 0%

去除引用文献复制比：1.7% **总文字复制比：1.7%**

单篇最大文字复制比：0.6%（无线传感器网络安全路由与密钥预分配技术研究与应用）

重复字数：	[858]	总字数：	[51124]	单篇最大重复字数：	[304]
总段落数：	[5]	前部重合字数：	[216]	疑似段落最大重合字数：	[330]
疑似段落数：	[4]	后部重合字数：	[642]	疑似段落最小重合字数：	[111]

指　标： ☐ 疑似剽窃观点 ☑ 疑似剽窃文字表述 ☐ 疑自我剽窃
☐ 一稿多投 ☐ 过度引用

表格：**0** 公式：**没有公式** 疑似文字的图片：0 脚注与尾注：**0**

2.6%	(304)	基于进化算法的FPGA脉动阵列快速硬核布局方法_第1部分 （总11742字）
0%	(0)	基于进化算法的FPGA脉动阵列快速硬核布局方法_第2部分 （总10433字）
1%	(111)	基于进化算法的FPGA脉动阵列快速硬核布局方法_第3部分 （总11134字）
1%	(113)	基于进化算法的FPGA脉动阵列快速硬核布局方法_第4部分 （总11425字）
5.2%	(330)	基于进化算法的FPGA脉动阵列快速硬核布局方法_第5部分 （总6390字）

（注释： 无问题部分 文字复制比部分 引用部分）

指导教师审查结果

指导教师：陈翔

审阅结果：
审阅意见：指导老师未填写审阅意见

1. 基于进化算法的FPGA脉动阵列快速硬核布局方法_第1部分

总字数：**11742**

相似文献列表

去除本人已发表文献复制比：2.6%(304) **文字复制比：**2.6%(304) **疑似剽窃观点（0）**

1	无线传感器网络安全路由与密钥预分配技术研究与应用 王杨子(导师：管有庆) -《南京邮电大学硕士论文》- 2019-12-09	2.6%（304） 是否引证：否
2	端到端网络切片动态资源优化算法研究 董旭(导师：赵力强) -《西安电子科技大学硕士论文》- 2019-09-01	2.0%（238） 是否引证：否
3	基于机器学习的定制化网络切片技术研究 周倩文(导师：赵力强) -《西安电子科技大学硕士论文》- 2019-06-01	2.0%（238） 是否引证：否
4	基于表面肌电和运动学分析的脑卒中下肢康复器械与相关诊疗方法研究 纪晴(导师：叶学松) -《浙江大学博士论文》- 2018-10-01	0.3%（39） 是否引证：否
5	压敏漆图像配准与修复方法研究 梁诚(导师：高志升;张海波) -《西华大学硕士论文》- 2019-04-01	0.3%（35） 是否引证：否
6	固件代码动态污点分析技术 任玉柱(导师：郭玉东) -《战略支援部队信息工程大学硕士论文》- 2019-10-15	0.3%（35） 是否引证：否
7	基于深度学习模型的图像检索研究 程毅玮(导师：冯兴杰) -《中国民航大学硕士论文》- 2019-05-11	0.3%（35） 是否引证：否
8	基于ARM的CNG加气站安全监管系统的设计与实现 王志亮(导师：官洪运) -《东华大学硕士论文》- 2011-01-01	0.3%（35） 是否引证：否

原文内容

本科毕业论 （设计）

题 : 基于进化算法的 FPGA 脉动阵列快速硬核布局法

院系: 电与信息程学院

专业: 通信程学姓名: 张年崧

学号: 16308149

指导教师: 陈翔教授, Nachiket Kapre 教授时间： 〇 〇年四七

基于进化算法的 FPGA 脉动阵列快速硬核布局法摘要

论文题目： 基于进化算法的 FPGA 脉动阵列快速硬核布局法专业： 通信程学生姓名： 张年崧学号： 16308149

指导教师： 陈翔教授, Nachiket Kapre 教授

摘要

现代端 FPGA 拥有计算密集度、丰富的异构硬核资源和专的速布线络，

为数字信号处理任务提供近似 ASIC 的计算性能。近年来，在性能 FPGA 上实现规模速神经络加速器在学术界和业界获得了极的关注，但此类规模设计在实现过程中临着布局和布线的难题。

向硬核利率的脉动阵列卷积神经络加速器设计需求，本提出种基于进化算法的快速硬核布局法，在运时间、线长、流线寄存器消耗成本、和时钟频率超越常的退算法和前最先进的解析算法。本主要有四创新点，其是利 Xilinx UltraScale+ FPGA 的 RAM 与 DSP 硬核设计了速卷积脉动阵列， 于卷积神经络的加速；其是将 FPGA 布局转化为有约束的多标优化问题，提出使配排序遗传算法 (NSGA-II) 和协差矩阵适应进化策略 (CMA-ES) 解决该问题，并设计了创新的基因型表 ; 其三是基于 Xilinx RapidWright FPGA 定制化框架搭建了动化、

端到端的布局布线具 RapidLayout，其运时间相 Vivado 快 5–6 倍；其四是设计了不同 FPGA 间的布局迁移学习法，加速同设计部署到不同 FPGA 的布局过程。

本针对上述所提案，对布局多个指标进量化并与 Versatile-Place-and-Route

(VPR), UTPlaceF 等最先进布局框架进对实验，通过数据结果和过程可视化说明了

本提出的布局法和实现框架已经达到了领先平。RapidLayout 的进化布局算法（1）

相 VPR 的退引擎最缩短 33% 的运时间，在线长（Wirelength）和边界框尺

(Bounding Box Size) 分别减了 1.9–2.4× 与 3.1–4.1×；（2）相于 UTPlaceF 解析布局算法，运时间最加快 9.2 倍，同时线长减了 1.8–2.2×，边界框尺减了 2–2.7×；

本提出布局结果在不同 FPGA 之间的迁移学习法，相于随机初始化的布局搜索运时间加快了 7-12 倍。

[关键词] FPGA 布局，卷积神经络加速器，优化算法，进化算法

I

RapidLayout: Fast Hard Block Placement of FPGA-optimized Systolic Arrays using Evolutionary Algorithms

ABSTRACT

Title: RapidLayout: Fast Hard Block Placement of FPGA-optimized Systolic

Arrays using Evolutionary Algorithms

Major: Telecommunication Engineering

Name: Niansong Zhang

Student ID: 16308149

Supervisor: Prof. Xiang Chen, Prof. Nachiket Kapre

Abstract

Modern high-end FPGAs are equipped with abundant heterogeneous computation resources with dedicated routing network, achieving near-ASIC computation performance and clock fre- quency. Large-scale neural-network accelerators on heterogeneous FPGAs have attracted re- search interest from academia and industry.

Evolutionary algorithms can outperform conventional placement algorithms such as sim- ulated annealing, analytical placement as well as analytical placement on metrics such as run- time, wirelength, pipelining cost, and clock frequency when mapping FPGA hard block in- tensive designs such as systolic arrays on Xilinx UltraScale + FPGAs. For certain hard-block intensive, systolic array accelerator designs, the commercial-grade Xilinx Vivado CAD tool is unable to provide a legal routing solution without tedious manual placement constraints. In- stead, we formulate an automatic FPGA placement algorithm for these hard blocks as a multi- objective optimization problem that targets wirelength squared and maximum bounding box size metrics. We build an end-to-end placement and routing flow called RapidLayout using the Xilinx RapidWright framework. RapidLayout runs 5–6× faster than Vivado with manual constraints, and eliminates the weeks long effort to manually generate placement constraints for the hard blocks. We also perform automated post-placement pipelining of the long wires inside each convolution block to target 650 MHz URAM-limited operation.

RapidLayout outperforms (1) the simulated annealer in VPR by 33% in runtime, 1.9-2.4× in wirelength, and 3.1-4.1× in bounding box size, while also (2) beating the analytical placer

UTPlaceF by 9.2× in runtime, 1.8-2.2× in wirelength, and 2-2.7× in bounding box size. We employ transfer learning from a base FPGA device to speed-up placement optimization for similar FPGA devices in the UltraScale + family by 7–12× than learning the placements from scratch.

[Keywords] FPGA Placement, Convolutional Neural Network Hardware Acceleration, Op- timization Theory, Evolutionary Algorithms

III

基于进化算法的 FPGA 脉动阵列快速硬核布局法录录

第 1 章引言

本章先介绍 FPGA 在智能发展背景下的应场景，以及规模计算密度

的设计对于 FPGA 实现过程的挑战，由此说明 FPGA 布局的选题背景，然后总结了已

有的 FPGA 布局法和研究现状，接着介绍本的主要作，最后概述本的结构与章节安排。

1.1 选题背景与意义英特尔的创始之登 · 摩尔提出，每隔 18-24 个单位积集成电路可容纳的元器件就会翻倍，其性能也将提升倍[1]，这条著名论断被们称为摩尔定律

（Moore’s Law）。然随着晶体管逐渐变，其静态功耗不断增加[2]，功耗上升和能量效率降低使得摩尔定律逐渐失效。在“后摩尔时代”，研究者和从业者们不再单纯追求计算密度的提，转寻求计算架构的创新。相对于 CPU 的复杂、低并度，与 GPU

的通、并度，FPGA 为芯设计者们提供了种领域定制架构[3](Domain-Specific

Architecture) 的全新思路：普通 FPGA 虽然频率较低，但具有并、可配置、低功耗、

能量效率的特点；开发周期短，相于 ASIC 更适于快速发展的神经络硬件计算领域，从降低重复性程成本。对 CPU 集中和复杂的内存管理式，FPGA 的存储为分布式且简单可配置。同时，现代 FPGA 多配备量专计算单元，可提供较规模的计算并度。现代的性能 FPGA 在保留其灵活性的同时可以实现近似 ASIC

的计算性能与时钟频率[4]，这得益于其异构计算资源的设计：除了数百万传统的查找表

(LUTs) 和寄存器 (Flip-Flops)，异构 FPGA 还在上引了 DSP，RAM 等硬核 IP 实现近存储计算[5](Near-Storage Computation)，这样的硬核具有计算效率 、存储容量 、时钟频率 、 持级联等特点[6]。同时，可级联硬核之间设计有专速布线资源，可构建速数据通道， 便数据快速搬运。然 ，数量庞的计算资源、分布和尺不规则

的异构计算存储 IP、逐渐增的计算架构规模，新架构带来的布局约束条件等等，都为FPGA 设计的布局与布线过程带来了挑战[6]。以 Xilinx Vivado Design Suite 在 UltraScale

与 UltraScale+ 系列 FPGA 上实现卷积神经络加速器为例，完整的实现流程需要耗费数时甚数天[7]，不仅需要耗费量进硬核布局设计，并且法保证实现结果可以达到标时序要求。因此， 向型异构计算密度 FPGA 的布局布线算法成为了近年来许多学者的研究重点。 1

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 1 章引

FPGA 设计的实现般包括 6 个步骤[8]：合成 (Synthesis)， 艺映射 (Technology

Mapping)，打包 (Clustering/Packing)，布局 (Placement)，布线（Routing）， 成特流

(Generate Bit-Stream)。FPGA 布局是把打包后的表对应到板上的物理元件位置上，布局的结果直接影响设计能否成功布线与设计的时序表现， 在实现的步骤中布局又是耗时较的步。因此，对 FPGA 布局的优化是提 CAD 具性能的关键。已有的主流应布局法主要分为三类，基于模拟退 (Simulated Annealing) 的布局算法[9]、解析布局算法 (Analytical Placement)[10]、和 Min-Cut 法[11]。模拟退法出现较早且应泛，经过随机初始化的布局在温度函数的控制下逐渐降低接受随机变化的概率，使系统冷却，从得到最优布局结果。退法得到的布局结果质量 ，但所需迭代优化时间较长，并度和可扩展性较差。Min-Cut 法将逻辑表不断按区域划分，直到表内元件完全展开和 FPGA 物理位置对应，适于型布局问题，且局部优化容易收敛局部最优。近年来获得较发展的法是解析布局法，其将 FPGA 布局优化问题建模为程系统，并使求解算法解得布局结果。解析布局算法可扩展性强，结果质量较 ，但需要代价较的布局结果合法化过程，可并度较差。

近年强化学习[12] 和神经络架构搜索算法[13;14] 的发展促进了进化算法的兴起，诸如经典的多标优化法 NSGA-II[15]（Non-Dominated Sorting Genetic Algorithm,

配排序遗传算法）和 State-of-the-Art 求导优化算法 CMA-ES[16]（Covariance Matrix

Adapted Evolution Strategy, 协差矩阵适应进化策略）。这些进化算法在许多多标、

箱、 平滑或连续域优化问题上取得了领先的表现， 且进化算法具有良好的可拓展性，其基于种群进化的优化式天然具备适应并计算的优势。因此，将新的进化算法应于 FPGA 布局优化问题有希望解决型异构 FPGA 的布局困难问题，具有极的可性与研究意义。

1.2 国内外研究现状和相关工作近年来，在 FPGA 上实现神经络加速器受到了研究者和从业者的泛关注，随着性能、低延迟的 FPGA 神经络加速器研究兴起， 规模的 FPGA 设计实现对 EDA

具提出了更的要求，吸引了国内外学者的研究兴趣。本节通过对种性能 FPGA

脉动阵列加速器的总结和基于进化算法的 FPGA 布局法发展从整体上阐述智能硬件加速与 EDA 领域的研究现状和相关作。 2

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 1 章引

1.2.1 基于 FPGA 的脉动阵列加速器脉动阵列的提出解决了硬件加速器随着规模增速度降低的问题。脉动阵列的结构是将速计算单元以 2D 的式排列并级联形成络，数据从端按节拍连续输 ，

经过运算的结果则从另端按照节拍连续输出。Google TPU 先将脉动阵列式的矩阵乘法加速器于机器学习计算的加速。脉动阵列不仅限于矩阵乘法，也可以于卷积神经络的计算加速，如 CLP[17;18]，Cascades[7]，和 Xilinx SuperTile[19;20]。

然实现利率的设计同时保持运算速度分具有挑战性。CLP[17;18] 基于

Virtex-7 系列 FPGA 实现，且没有使上 DSP 计算资源，其时钟频率仅为 100-170

MHz。Xilinx SuperTile[19;20] 在 UltraScale+ FPGA 上实现，时钟频率可以达到 750 MHz，

但与此同时放弃了 URAM 的带宽， 且浪费了 50% 的上 DSP 资源。Cascades[7] 同样基于 UltraScale+ FPGA 实现，其时钟频率达到 650 MHz，DSP、 上 RAM 的利率均达到 95%， 且没有限制 URAM 的带宽。但是， 利率的脉动阵列设计带来了布局上的困难：Cascade[7] 设计法依赖 Vivado 内置的布局引擎产可布线的结果， 必须设计 480 个卷积单元内部硬核的布局位置，才可以成功布局，并达到 URAM 上限的 650 MHz 时钟频率。

1.2.2 基于进化算法的 FPGA 布局方法将进化算法应到 FPGA 布局优化曾有过许多尝试，但很少获得成功。最早的尝试由 Venkatraman 与 Patnaik[21] 提出，这项作将 FPGA 上的元件位置维坐标编码为遗传算法的基因型，并且使基于关键路径长度估计和空间利率的标函数进优化。

这项作为 FPGA 布局法提供了新思路，但并未取得传统的模拟退法更优的布局结果。近年来，P. Jamieson[22;23] 指出遗传算法法取得与模拟退布局算法相当的结果质量是由于 crossover 算在探索解空间的能上有限。因此提出了 Supergenes[24] 作为种改进的遗传算法。这项作引了分组算克服 crossover 算的缺点，并产了和退相当质量的结果。Praveen T. 对遗传算法进了平化[25]，并将遗传算法和退布局法结合，利退法提布局结果的质量， 遗传算法对布局搜索过程加速。

总体 ，将进化算法应于 FPGA 布局经历过较多尝试，但并未获得相模拟退法更优的性能。然 ，近年来强化学习和神经络架构搜索的发展推动了新的进化算法发展：NSGA-II 在强化学习的解空间探索上展了良好性能[12;13]，CMA-ES 作为种求导的优化法在神经络架构搜索上得到了应 [14]。并且进化算法基于种群的优化思想具有对平计算加速的良好持，且不需要代价的解合法化过程。因此，

将更先进的进化算法应于 FPGA 布局问题具有较研究意义。 3

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 1 章引

1.3 本文的研究内容与主要工作本的研究内容是通过进化算法对计算密度的卷积神经络加速器进布局优化，并设计全动、端到端的 FPGA 布局布线实现框架。由于计算密度的神经络脉动阵列加速器法依靠现有的布局算法得到可的全局布局结果，本提出对此类加速器设计进硬核布局优化，并探索通过布局结果的复幅降低整体设计实现时间的法。

本的作是将 FPGA 脉动阵列硬核布局问题建模为受限的多标优化问题，将配排序遗传算法 (NSGA-II) 和协差矩阵适应进化策略 (CMA-ES) 应于布局问题的求解，构建布局实现框架，解决了硬核利率设计的布局困难问题，消除了

布局硬核的烦，且得到了超越退和其他最先进布局算法的性能与 5–6 倍相对于Vivado EDA 作流的加速。本的主要作分为以下点：

1) 回顾和总结了经典的脉动阵列加速器设计，指出加速器计算密度增和时钟频

率的要求对 EDA 中的布局问题的挑战, 并总结已有的 FPGA 布局算法并较优劣，回顾已有对进化算法布局的尝试；

2) 将异构 FPGA 的硬核布局建模为多标优化问题，应 NSGA-II 与 CMA-ES 两

种进化算法求解。针对脉动阵列的级联规则造成的布局限制，提出种创新的基因型设计，将布局问题划分为三个问题并将布局限制编码进优化过程，量化布局结果质量指标，并与标准布局具 VPR 和最先进的解析布局算法 UTPlaceF

对 ，评价进化算法的性能；

3) 设计和实现了动化、端到端的快速 FPGA 布局布线框架，RapidLayout，并与

Vivado 布局作流对 ，评价实现效率的提升；

4) 为硬核布局结果跨设备优化过程加速设计了布局的迁移学习法，将布局案迁

移到 UltraScale+ VU3P 到 VU13P 的全系列 FPGA，评价加速指标。

1.4 本文的论文结构与章节安排本共分为五章，各章节内容安排如下：

第章为引，阐述了基于进化算法的 FPGA 脉动阵列布局法选题的研究背景和意义，对国内外研究现状和相关作的概述，以及本的主要创新点及研究内容，最终介绍本的章节安排。

第章是相关技术法概述，简要介绍了基于进化算法的 FPGA 脉动阵列布局所涉及的各技术。 先总结了经典的神经络硬件加速器设计，然后介绍了 FPGA 的 4

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 1 章引组成元件、EDA 实现过程，并介绍了 RapidWright FPGA 定制化实现框架，最终总结了四类 FPGA 布局法。

第三章是本的重点内容， 先明确布局问题的范围并建模为多标优化问题。其

次，详细阐述了基于进化算法硬核布局的基因型设计，并明确了 NSGA-II 与 CMA-ES

的实现原理。在问题建的基础上，本重点介绍了 RapidLayout FPGA 框架的实现细

节，最后层层递进地以 VU11P FPGA 的布局实现为例说明了 RapidLayout 的作流程。

第四章是实验分析与结论， 先介绍了实验中使的 FPGA 器件和实验中涉及的

对布局算法，其次描述了实验的编程实现与测试环境等实现细节，然后以多种法呈现布局法性能与收敛过程的对结果并加以分析，最后展了 RapidLayout 的迁移学

习性能，说明了 RapidLayout 对其他 FPGA 的适性。

第五章是总结与展望，总结了本的内容，分析所提出法与框架的优点与不 ，

然后进未来展望，说明此项作未来的发展向。 5

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 2 章神经络硬件加速相关技术概述

第 2 章神经网络硬件加速相关技术概述

本章就基于进化算法的 FPGA 脉动阵列布局所涉及的各技术进总结与阐述。

先总结经典的神经络硬件加速设计，主要涉及基于线性缓存的卷积核设计与速脉动阵列。然后介绍 FPGA 的基本实现原理与 RapidWright FPGA 定制化实现框架，最后介绍经典的类 FPGA 布局算法与基本的进化算法。

2.1 神经网络硬件加速随着智能的兴起，卷积神经络在计算机视觉领域得到了规模应， 如图像分类[26]、物体检测[27]、语义分割[28] 等诸多领域。同时，因为卷积神经络训练和预测中的计算需求和易于并化的特点，在可编程硬件上实现神经络计算架构受到了泛的研究关注。本节主要介绍两种经典的卷积神经络加速器设计：Line-Buffer

型卷积计算单元[29] 与脉动阵列[7]。

图 2-1 Line Buffer 型卷积计算单元[29]

如图2-1所为 Line-Buffer 型卷积计算单元的基本结构，其特点是使移位寄存器构成数据缓存器，通过不断输像素数据完成等效的卷积滑窗操作。数据缓存器窗内的像素值与权重缓存器内的权重值并输乘法器阵列。乘法器阵列的结果经过加法树 7

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 2 章神经络硬件加速相关技术概述并累加得到次卷积操作的结果。Line-Buffer 型卷积计算单元的设计并度为卷积核尺 ，优点是结构简单、并度较 、占资源少，但基于 LUT 和 Flip-Flop 实现会导致其时钟频率较低，因此更适合在边缘端或移动端进前向计算。

指　标

疑似剽窃文字表述

1. 1.4 本文的论文结构与章节安排本共分为五章，各章节内容安排如下： 第章为引
2. 选题的研究背景和意义，对国内外研究现状和相关作的概述，以及本的主要创新点及研究内容，最终介绍本的

2. 基于进化算法的FPGA脉动阵列快速硬核布局方法_第2部分

总字数：**10433**

相似文献列表

去除本人已发表文献复制比：0%(0)　　**文字复制比：**0%(0)　　**疑似剽窃观点（0）**

原文内容

第种经典卷积神经络加速架构是脉动阵列。脉动阵列的提出是为了解决规模加速器难以实现速运算的问题。脉动阵列以许多乘法器或其他运算单元在横向和纵向级联组成规模计算阵列，数据连续从端按节拍输 ，计算结果将从另端连续

按节拍输出，因此称作脉动阵列。脉动阵列不仅适于矩阵乘法，也适于卷积计算。

DSP48 DSP48 DSP48 DSP48 DSP48 DSP48 DSP48 DSP48DSP48

RAMB 18

RAMB 18

RAMB 18

URAM 288

URAM 288 +

DSP48 DSP48 DSP48 DSP48 DSP48 DSP48 DSP48 DSP48DSP48

RAMB 18

RAMB 18

RAMB 18 72 8 8 8 8 16

24RAMB 18

RAMB 18 8 8 8 8 8 8 16 24

图 2-2 脉动阵列卷积计算单元[7]

图 2-3 卷积运算单元和矩阵乘法运算单元级联的脉动阵列[7] 8

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 2 章神经络硬件加速相关技术概述图2-2为 Scale and Cascade[7] 提出的脉动阵列卷积计算单元架构图。 个卷积计算单元由 2 个 Ultra RAM，8 个 Block RAM，18 个 DSP48 构成。其中输 URAM 于存储特征图的两个通道，和个卷积核的权重。URAM 连接着两组乘加计算单元，每组分别有 3 个 BRAM 负责移位寄存特征图像素，1 个 BRAM 存储卷积核权重。在乘加计算单元权重和特征像素不断由 BRAM 输送给级联的 DSP48，在这计算乘积并将结果累加。最后个 DSP 输出的值就是次卷积的计算结果，并由输出端的 URAM 存

储。此设计的并度为 2 倍卷积核数，但特征图像素数据复更 ，同时时钟频率相Line-Buffer 设计更 ，更适合规模神经络加速。

图2-3说明了卷积单元和矩阵乘法单元级联组成脉动阵列的式。矩阵乘法单元的

设计与卷积类似，不同之处在于乘加单元中 BRAM 数增加到 9 个，计算结果由个BRAM 存储。URAM 的级联同时增加了特征图的通道复 ，进步提了设计的并度。

2.2 FPGA EDA 实现原理

FPGA 设计的 EDA 实现过程主要分为 5 个部分[8]：合成 (Synthesis)， 艺映射 (Tech- nology Mapping)，打包 (Packing)，布局 (Placement)，布线 (Routing)。本节分别对每个步骤进阐述。

图 2-4 FPGA 设计的 EDA 实现步骤[8]

合成和艺映射：在这两个步骤中，电路设计编译和转化为逻辑表， 表中通常包括信号线与 LUT 和 Flip-Flop。

打包：此步骤将上步产的表进打包，将 LUTs 和 Flip-Flop 打包成 CLB(Configurable

Logic Block)。经过打包阶段的逻辑表可以与 FPGA 上的计算单元相对应，可 9

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 2 章神经络硬件加速相关技术概述于后续的布局布线。

布局: 布局阶段为打包之后的逻辑表中的逻辑单元确定对应的物理位置，同时优化某些标， 如最化阻塞、线长等， 的是在于产可以布线的布局结果，

同时尽可能满时序要求。

布线：将布局之后的电路根据信号线连接起来。在 FPGA 中横向或纵向的连接是

通过 PIP（Programmable Interconnect Points）完成的， 横向与纵向的连接通过 switch box 完成。这阶段也包括时钟信号的连线。经过布线的设计可以产时序报告，得到最运时钟频率。

2.3 RapidWright FPGA 实现框架图 2-5 RapidWright FPGA 实现框架[30] 10

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 2 章神经络硬件加速相关技术概述

2.3.1 RapidWright 简介RapidWright[

30]是 Xilinx Research Lab 开发的开源 Java FPGA 定义框架，能够修

改基于 Xilinx FPGA 和 SoC 的逻辑表设计与物理实现修改，它提供了 Xilinx Vivado

Design Suite 许多功能的补充，其中包括：

快速加载和浏览 Vivado 持的 Xilinx 设备物理模型；

导和写出未加密的 DCP checkpoint 件；

精确修改 FPGA 设计的逻辑表和物理实现，如修改布局和布线；

提供种模块化的快速实现法，可将先前实现的设计作为新设计中的个模块，

对其修改、复制或迁移位置。

RapidWright 的优势在于它与 Vivado 的完全兼容性，对 Xilinx FPGA 设备物理信息的快速获取能和精确修改设计的能 。其基于 Java 的实现为许多创新性任务带来了便捷， 如通过 RapidWright 可以精确获得 FPGA 器件上某个元件的位置、它在设计中的状态：是否被于布局，是否完成布线等信息，甚具体到每个可编程连接点 (PIP)

和信号线的状态都可精确获知。RapidWright 持 Vivado TCL Interface 的全部功能，同时提供更的操作精确度和更便捷的获取法。

2.3.2 DCP 简介图2-5为 Vivado 与 RapidWright 的交互式。在 Vivado 合成到布线中间的每个过程中，都可将设计导出为个 DCP 格式的 checkpoint 件，DCP 件中包含了该设计的逻辑表、约束件、物理表实现结果与相应器件信息，可以完全确定 FPGA 设计

在该阶段的状态，因此 DCP 件也包含设计的布局与布线信息。

RapidWright 提供了快速的 DCP 件读和写出功能， 个完整的 FPGA 设计包括其逻辑表、约束和布局布线信息都反映在 RapidWright 的 Design 类中，在此基础上

可以对其中的每项通过 RapidWright 提供的 API 进精确的修改。修改完毕的设计使Design 类中的 DCP 写出 API 将修改后的设计导出为 Vivado 兼容的 DCP 格式，可

以 Vivado 打开、查看、继续修改， 成特流。RapidWright 提供了种向对象的FPGA 实现法，实现的所有信息都有对应的类表 ，在操作时使该类的对象以及提供的 API。 11

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 2 章神经络硬件加速相关技术概述

2.4 FPGA 计算架构构成FPGA(Field Programmable Gate Array，场可编程门阵列) 是类可以通过编程法修改其内部逻辑的芯 。早期的 FPGA 多包含由 SRAM 实现的可编程逻辑模块和 IO

模块，每个可编程逻辑模块中包含查找表和寄存器。 现代 FPGA 除了可编程模块加了多种新型 IP 核， 如上分布式 RAM，DSP 模块，PCIe 模块等等。本节先介绍 FPGA 的种基本组成元件，然后以 Xilinx FPGA 为例，介绍现代异构 FPGA 的组成结构，并详细介绍 UltraScale+ 系列配备的 IP 硬核[31]。

2.4.1 FPGA 的基本组成元件

LUT

LUT（查找表）是 FPGA 可编程逻辑的核 ，也是计算的基本组成单元。LUT

般是多输 、单输出的，输出信号通过输选择信号和存储器的内容确定。 前常的LUT 般是 6 个输信号。

图 2-6 FPGA 的基本组成元件：查找表 LUT 与真值表的关系[30]

图2-6展了个三输 LUT 的例 。LUT 通常以 N：1 选择器和个 N 特存储器的形式实现，在图2-6的例中 N=8，如果输特数为 k，则 N = 2k。LUT 能够实现任意 k 输单输出的逻辑函数，这是因为 LUT 与真值表的对应关系。输信号于多路选择器的信号选择， 真值表对应输的输出值则通过存储器存在多路选择器的对应位置中，这样每次仅需修改存储器中的值就可实现任意逻辑函数，这也就是 FPGA

计算可编程性的来源。

State Element

LUT 的计算值输出后通常需要寄存。因此多数的 FPGA 将 LUT 和 D-Flip Flop 连接为组逻辑，称为状态单元(State Element)。通常来说这的寄存单元使 reset/clear 12

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 2 章神经络硬件加速相关技术概述信号和时钟来确定作式，并可以作为锁存器使 ，这些状态单元通常都有专的时钟路径。FPGA 将任意数量的状态单元连接起来，就可以实现任意的逻辑函数。Xilinx

提供种基于 LUT 的状态单元变种， 持在 LUT 的存储器单元内寄存数据，这样的

LUT 可以通过编程当做型存储、移位寄存器或 FIFO 使 。

Carry Chain

Carry Chain 是 FPGA 的基本组成元件之 ，通常组 LUT 会配备个 Carry Chain

来实现效的可编程逻辑。Carry Chain 般于简单逻辑操作的专进位通道， 如加法、减法、 较等等。虽然 Carry Chain 的功能可 LUT 实现，但专的进位通道对

资源的利更加效，性能也更好。

DSP

在 FPGA 上直接 LUT 实现乘法器是常昂贵的，同时乘法又是极为普遍的运算之 ，因此许多现代 FPGA 专门设计了乘法计算单元。这些乘法计算单元般是专的 IP 硬核， 持整数乘法操作。随着 FPGA 的进化更先进的 DSP 也持乘累加操作和

宽位 AND/XOR 操作等等。

Block RAMs

现代 FPGA 通常在上设置型的存储器，可以达到 KB。这些存储器在

上通常是分布式的，并且使常灵活， 持单/双读写，拆分或者级联。以 Xilinx

UltraScale+ 架构为例，此系列 FPGA 提供两类 Block RAM， 类是 BRAM，另类是更的 URAM。

2.4.2 现代异构 FPGA 的组成架构本结以 Xilinx 架构为例阐述现代 FPGA 的组成结构。对于 Xilinx FPGA 来说，其架构主要分为 6 级，从最基本的可编程单元直到整个 FPGA。图2-7描述了该 6 个级

别。BEL: Basic Element of Logic

BEL 是 Xilinx FPGA 最的物理计算单元。BEL 分为两类， 类是 Logic BEL，另类是 Routing BEL。Logic BEL 实际上就是包含 LUT，FF 等在内的最逻辑计算单

元，实现逻辑计算。将逻辑表中的叶节点对应到 Logic BEL 上，就是“布局”的含义。

Routing BEL 是可编程的多路选择器， 来编程连通或断开 Logic BEL 之间的连接，也

就是布线。以 Routing BEL 为基础的布线资源称为 PIP，Programmable Interconnect Point。

Site

组有连接关系的 BEL 聚集在起构成了个 Site，在 Site 中有三种元件：BEL，

Site Pin 和 Site Wire。Site Pin 于连接内部元件和外部资源，也就是引脚。Site Wire 负 13

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 2 章神经络硬件加速相关技术概述图 2-7 Xilinx FPGA 可编程硬件组成结构[30]

责连接内部各个 BEL 和 Site Pin。在 Xilinx FPGA 架构中，每个 Site 有的位置编

号，不同种类的 Site 单独编号，因此可以便地找出其相对位置或于索引。

Tile

Tile 是构建 FPGA 的模块， 个 Tile 可包含多个 Site，但也可以不包含任何 Site。

Tile 内部和之间的连线是通过 PIP 实现的 Node 完成的。Node 是延伸多个 Tile 的电路

连接，从 Node 的连接状态可以获知其初始点与终点，以及连接中间经过的 Tile。Xilinx

FPGA 使种列主导的 Tile 构建式，也就是同列的 Tile 般是相同类型的。

FSR: Fabric Sub Region

FSR 就是通常所说的时钟域，其中包含多个 Tile，各有独的时钟资源。在 Xilinx

UltraScale 架构中，FSR 的度都相同， 宽度根据不同的内部构成有所不同。

SLR: Super Logic Region

SLR 也称为超逻辑域，其出现始于 FPGA 2.5D Packaging 艺的发展。现代型FPGA 般包含多个硅 ，每个单独的硅就是个 SLR，可看作个完整的独芯 。SLR 般包含多个时钟域，其边缘与其他 SLR 的连接通过专的跨逻辑域通路，

并有专的 Tile 辅助跨逻辑连接，叫做 Laguna Tiles[31].

Device

Xilinx FPGA 架构中最级别的架构是 Device，代表整个 FPGA， 般由个或多个 SLR 纵向堆叠构成。 14

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 2 章神经络硬件加速相关技术概述

2.4.3 UltraScale+ IP 硬核介绍Xilinx UltraScale+ 系列 FPGA 集成了上千个 IP 硬核 (hard block)，这些硬核具有时钟频率， 性能，可配置等特点，本结详细阐述三种 IP 硬核的特点。

图 2-8 Xilinx UltraScale+ IP 硬核介绍[7]

DSP48E2 （图2-8 a） 持 27 × 18 整数乘法，48 位累加。Xilinx DSP48 持的最时钟频率为 891 MHz，并且持功能选择。DSP48 不仅可以配置运算类型，还可以改变数据通路。因此，本的脉动阵列设计利 DSP48E2 可选数据通路和持级联的特性将 DSP48E2 设置为 MAC(Multiply and Accumulate) 功能，并且串流

输数据进复 。

RAMB18 （图2-8 b) Xilinx FPGA 集成数千个分布式块存储器，其输输出位宽

可以配置。RAMB18 最持时钟频率为 825 MHz，并且提供特殊的级联配置。

RAMB18 允许于将邻近的多个 BRAM 级联成为个更深的存储器或者 FIFO。

本利 RAMB18 的级联特性进卷积设计中的特征图数据复 。

URAM288 （图2-8 c) UltraScale+ 独有的超型上存储器 Ultra RAM，也就是

URAM288.URAM288 持 288kb 存储，其位宽不可配置，但仍持级联配置。

URAM288 持的最时钟频率为 650 MHz。在实际布局问题中，本的优化标就是使布线后的完整脉动阵列加速器时钟频率不低于 650 MHz。

2.5 FPGA 布局算法FPGA 布局问题是将打包后的逻辑表中的元件对应到 FPGA 物理位置的过程，其标是寻找对应关系的同时优化特定标， 如设计可布线性，关键路径长度或阻塞程度。FPGA 布局已被证实为 NP-complete 问题[32]，且 FPGA 的布局结果直接影响设计是否可布线以及实现后的时序表现，因此布局是 EDA 问题中的重点和难点。本节通过总结已有的和近年发展的 FPGA 布局法从整体度阐述 FPGA 布局算法原理。FPGA 布

局算法可基本分为四类：基于模拟退的布局算法[9;33]，进化类算法[21;24;34;35]，Min-Cut

Partition[

36]，解析类布局算法[37;38]。 15

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 2 章神经络硬件加速相关技术概述

2.5.1 模拟退火模拟退法出现较早且应泛，经过随机初始化的布局在温度函数的控制下逐渐降低接受随机变化的概率，使系统冷却，从得到最优布局结果。基于模拟退法的布局具 VPR 在学术界获得认可，其改进的标函数泛适于各种架构和电路设计布局。退法得到的布局结果质量 ，但所需迭代优化时间较长，并度和可扩展性较差。另外，退法的结果受其温度设定影响较 。退的温度函数也称为冷却进度表，

其规定退的初始温度，下降速率等信息。冷却过快会导致系统法充分收敛，最终的优化结果较差；冷却过慢则导致收敛时间过长， 法在可时间范围内给出最优解。因此，退法需要调节参数，收敛速度与结果质量与具体优化设置度相关。

2.5.2 进化算法进化算法是类受到达尔进化论启发的优化算法，经典的例包括遗传算法、蚁群算法、粒群算法等等[39]。进化算法作为种迭代优化法其核思想在于使基因型编码可解，定义标函数（或称为适应度函数），然后通过基因型的交叉、变异在迭代过程中产新的可解，然后通过标函数将适应度低的基因型或个体淘汰，以此使种群趋向于标函数的向进化。 般认为，应于 FPGA 布局的进化算法表现不如退 ，其原因在于交叉算对解空间探索性能较差。但进化算法基于种群进化的思想具有适应并化的优势，因为同迭代周期内的个体可以同时计算适应度，不存在依赖关系； 退下次随机操作基于当前的状态，因此不具有并化的优势。 且，

近年来智能和优化领域的发展推动了多标和鲁棒性进化算法的发展，并已在

强化学习、神经络结构搜索等领域取得了领先的表现。因此，将现代进化算法应于FPGA 布局是具有前景的研究向。

2.5.3 Min-Cut Partition

Min-Cut 法将逻辑表不断按区域划分，直到表内元件完全展开和 FPGA 物理位置对应。逻辑表被不断划分为两个表，同时，芯的物理位置也在横向和纵向不断被划分为区域，每个表被指定在个对应的区域内。这过程不断重复，直到每个区域内仅含有个逻辑元件为 。Min-Cut 法适于型布局问题，且局部优化容易收敛局部最优[40]。 16

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 2 章神经络硬件加速相关技术概述

2.5.4 解析布局算法近年来获得较发展的法是解析布局法，其将 FPGA 布局优化问题建模为程系统，并使求解算法解得布局结果。常的求解算法如 QP 法 (Quadratic Programming),

通过求梯度求解标函数的最值。解析布局算法可扩展性强，结果质量较 ，但需要代价较的布局结果合法化过程，可并度较差。 前泛应的 Vivado Design Suite

中使的布局算法为多标解析算法，优化标为时钟频率，阻塞和线长[41]。

2.6 本章小结本章先介绍了神经络硬件加速的两种经典架构，然后阐述了 FPGA EDA 实现过程的基本步骤，明确了 EDA 中布局过程的重要性和挑战。接着详细介绍了本使的 FPGA 实现具 RapidWright 与 Xilinx FPGA 的计算架构。最后，本章介绍了前主流的四类 FPGA 布局算法，分别阐述了各的基本原理、总结优势以及不之处。 17

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 3 章基于进化算法的 FPGA 脉动阵列快速硬核布局法

第 3 章基于进化算法的 FPGA 脉动阵列快速硬核布局方法

本章详细介绍本的布局问题建模，NSGA-II 和 CMA-ES 的算法原理，RapidLayout

框架流程与实现细节。 先介绍了向 UltraScale+ FPGA 的布局问题建模，明确了问题的优化标与约束；其次讨论了本的创新基因型设计与相应的解合法化过程；然后阐述了 NSGA-II 与 CMA-ES 的算法原理与实现案；最终介绍了 RapidLayout 布局框架的作流，并以层层递进的式举例阐述了脉动阵列的布局实现过程。

3.1 问题建模

本将 FPGA 硬核布局建模为个存在约束的多标优化问题。 先，本以 Xilinx

UltraScale+ VU11P FPGA 举例说明异构 FPGA 中硬核的排布法和架构带来的布局约

束条件，建布局问题的背景；然后说明本中布局问题的多个优化标，以及根据优

化标和约束条件建布局模型的法。

图3-1展了 Xilinx UltraScale+ 系列 VU11P FPGA 的整体架构和放后两个时钟域内部的硬核排布。 先，VU11P FPGA 由 3 个超逻辑域 (Super Logic Region, SLR) 构成，每个超逻辑域内的可编程器件构成和排布完全相同；其次，超逻辑域内包含三种硬核，分别是 DSP48E2, RAMB18, URAM288，每种硬核按列排布，但不同硬核列之间

的分布却是规则的；同时，三种硬核的尺不同：URAM288 最 ，RAMB18 其次，

DSP48E2 最 。VU11P 共有 960 个 URAM288 单元、4032 个 RAMB18 单元，9216

个 DSP48 单元，因此硬核的资源是不均衡的。硬核列之间排布的不规则、硬核资源的不均衡和硬核本的不同给布局带来了挑战：低质量的排布会造成部分计算单元内部连线过长，导致整体设计时钟频率低，且可布线程度 (Routability) 降低导致布线失败。另 ，由于硬核之间为级联设计了专的速布线资源，所以硬核的布局存在约束：

（1）级联的硬核必须按从下 (South) 到上 (North) 的顺序排布，且不能跨超逻辑域

（SLR）；（2）级联的 DSP48E2 与 URAM288 必须映射在相邻的两个 Site 上；（3）级联

的 RAMB18 必须映射在距离 1 个 Site 的相邻两个位置上；

FPGA 布局的的般有两个：将 Technology Mapping 并分组后的逻辑表映射到FPGA 的物理器件上，保证设计可以布线；另个的是在可布线的基础上追求的延迟 (delay)，也就是尽可能短的关键路径 (Critical Path) 和尽可能的时钟频率。因此，我们将硬核布局问题建为个有约束的多标优化问题： 19

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 3 章基于进化算法的 FPGA 脉动阵列快速硬核布局法图 3-1 Xilinx UltraScale+ VU11P 硬核列分布意图本将卷积运算单元 k 记为 Ck，逻辑表中硬核 i 对应的物理位置横纵坐标分别

记为 xi，y i

， xi

,j

与 yi

,j

分别表逻辑表中硬核 i 与硬核 j 布局后物理位置横坐标与纵坐标的绝对值之差。XMAX，Y MAX 表布局范围物理边界，即物理位置横纵

坐标的最值。

Minimize: ∑ i,j

((xi

,j+ y i,j) · w i,j)2(3.1) max k

(BBoxSize(Ck)) (3.2) subject to:

0 ≤ xi< XMAX, 0 ≤ y i< Y MAX (3.3) xi, y i = x j, y j(3.4) 20

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 3 章基于进化算法的 FPGA 脉动阵列快速硬核布局法 if i is cascaded after j in the same column: xi= x j yi= yj+ 1 i, j ∈ {DSP, URAM } yj+ 2 i, j ∈ {RAMB} (3.5)

符号说明： i ∈ {DSP, RAM, URAM} 表逻辑表映射到的 FPGA 上物理硬核单元。

Ck表卷积计算单元 k，其中包含 2 个 URAM，18 个 DSP 和 8 个 BRAM。 xi

,j+ y i,j

表硬核 i 与硬核 j 之间的曼哈顿距离 (Manhattan Distance)。 wi

,j

表硬核 i 与硬核 j 之间距离的权重，这我们两个硬核之间连线的数

作为权重。

BBoxSize() 表卷积核 Ck的 Bounding Box 边界框尺 ，也就是边界框宽与

的和. xiand y i

表硬核 i 的 RPM 绝对坐标[31]， 于计算两硬核之间的距离和卷积计算单元的边界框尺 .

3.1.1 布局问题的目标函数

我们线长的平来表征布局的阻塞性能，如式（3.1）， 最边界框（Bounding

Box）尺来表征卷积单元的关键路径（Critical Path）如式（3.2）。这两个标函数是紧密相关的，他们的的在于同时尝试降低流线寄存器消耗成本和提时钟频率。在实验中我们观察到，仅优化线长的平会导致个别卷积运算单元收敛到横向长条形布局结果，导致关键路径出现在控制逻辑，时钟频率低且法做流线；仅优化最边界框尺 ，则会导致优化过程常不稳定，容易收敛到局部最值。因此，我们选择结合这两个标函数来约束布局资源消耗同时追求更的时钟频率。

3.1.2 布局问题的约束条件优化器仅需要遵守 3 个约束条件。（1）式（3.3）所表的区域约束条件。它规定了布局优化所使的硬核必须在 FPGA 上 XMAX ×Y MAX 的矩形区域内。（2）式（3.4）所表的非重合条件。它来防优化器将多个逻辑硬核映射到同个物理位置上。

3. 基于进化算法的FPGA脉动阵列快速硬核布局方法_第3部分

总字数：**11134**

相似文献列表

去除本人已发表文献复制比：1%(111) **文字复制比：**1%(111) **疑似剽窃观点（0）**

1	基于CMAES算法的加速度计多位置新型标定方法 仲志丹;张玮琪;杜慧颖; -《智能计算机与应用》- 2018-04-28	1.0%（111） 是否引证：否

原文内容

（3）

式（3.5）所表的级联条件，它规定了级联的硬核必须遵守 Xilinx UltraScale+ 器件的布线规则。对于 DSP 和 URAM，级联的两个硬核必须从下到上映射在两个相邻的物理位置上。对于 BRAM 来说，级联的两个硬核必须从下到上映射到间隔位的两个相邻 21

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 3 章基于进化算法的 FPGA 脉动阵列快速硬核布局法物理位置上。BRAM 映射规则的特殊性是由 BRAM 的类型决定的：在 UltraScale 家族的器件中，块 RAM (Block RAM) 的基本类型是 RAMB36，其内部包含两个 RAMB18，

分别是 RAMB180 和 RAMB181。两种 RAMB18 都可以使 ，但是 RAMB180 仅能和RAMB180 级联，RAMB181 也仅能和 RAMB181 级联，这就造成了级联的 RAMB18 必须间隔位放置。

3.2 基于进化算法硬核布局的基因型设计

3.2.1 进化算法的基因型设计

Hard Block Placement Genotype

Distribution Location Mapping

DSP48 BRAM URAM … 4.1 6.3 5.2 7.6 4.3 … … … … 5.3 6.1 7.5 4.2 7.6 … … … … 4.3 6.5 3.7 6.2 5.6 … … …

(a) distribution

DSP48 BRAM URAM 0.3 .56 .67 7.6 4.3 …… … 0.0 0.3 0.6 0.7 0.8 …… … 0.9 .66 0.4 .35 0.1 …

……

(b) location 3 0 1 0 1 2 3 2 0 1 5 4 6 8 7

DSP48 BRAM URAM …0 2 1 3 … …… …0 3 1 2 … …… 1 0 … … … ……

(c) mapping

图 3-2 本提出的“三段”式硬核布局基因型设计。(a) Distribution 限定了每列中选定于布局的硬核个数；(b) Location 限定每个硬核在该列中的相对位置；(c) Mapping 规定了硬核之间的连接关系，也就是逻辑表中的硬核与 FPGA 上物理位置的对应关系。

对于以上讨论的布局问题，K 个卷积运算单元暴搜索法需要尝试的问题空间在 SP! × BRAM! × URAM! = 9K! × 4K! × 1K! 的数量级，显然是不通的。本提出基于进化算法的应和布局，可以在短时间内发现质量的布局结果。在进化算法中使基因型来编码可解，因此合理的基因型设计对优化效果有关重要的影响。我们将布局问题分解为三个问题：硬核的列分布，硬核在列中的位置，硬核之间的连接关系。根据这三个问题，我们设计了对应的基因型。

分布（Distribution）： 因为卷积加速器设计中使的各硬核例不定完全符合 FPGA 板上的硬核资源例，又由于 UltraScale 系列硬件的硬核资源为按列分布，因此将使的硬核资源按例分配到各列，这部分基因型就是规定了按什么样的例将使的硬核资源分配到各列。在解码该部分基因型时， 先将其归化，然后根据归化后的值和各列含有的硬核数决定每列放置多少个该种 22

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 3 章基于进化算法的 FPGA 脉动阵列快速硬核布局法类硬核。

位置（Location）：在我们决定了每列放置硬核的数之后，下步需要决定的是硬核在每列中的位置。这部分基因型规定的就是每个硬核在该列的位置。

此基因型使 0→1 浮点数对应硬核在列中从下到上的相对位置，也就是基因型中位置值接近 0 的硬核将被放置在列下的位置，接近 1 的硬核将被放在列上的位置。

对应关系（Mapping）：最终，当确定了硬核放置的数和位置之后，我们也就确定了在所有的物理硬核位置中，哪些硬核位置被选中于布局。最终需要确定的，

是将逻辑表中的硬核映射到物理位置时候的对应关系。我们将选中的硬核位置编号，对应关系基因型中规定这些位置编号的排列，在解码时按顺序将基因型中的排列对应到表中的硬核，这样就规定了逻辑表中硬核到物理硬核的对应关系。这实际上影响的是每个卷积单元中硬核的连接关系。

以上说明的三种基因型分别对应着布局问题的三个问题，在图3-2中，我们对

基因型设计进了可视化。这三种基因型是紧密相关、互相依赖的，因此法拆解为三个单独的问题分别优化。我们针对这特性应复合基因型[42] (Composite Genotype)

来适应问题的依赖关系，具体来说，三个问题分别编码，在优化过程中分别更新，

彼此之间不相互影响，但是在计算基因型的适应度函数 (Fitness Function) 的时候，三个基因型同解码计算。

3.2.2 基因型解码时的解合法化为了适应浮点优化法，我们通过解合法化 (Solution Legalization) 步骤对基因型采取量化等操作得到实际的布局，这步骤在解码时进 。算法 3.1 展了了分布基因型的解码和合法化过程。由于假定可资源已知，因此我们先将分布量化为整数，然后将每个整数值约束到每列的最可放置硬核数。然后，我们将上步去掉的硬核均匀分给其他的列。

算法 3.2 展了位置基因型的解码与合法化过程。由于硬块位置是离散的，因此将对基因型中的连续值进量化。每个量化后的值都是级联链中第个硬核的位置，所以从位置基因型直接解码的级联链位置可能会重叠或超出列中可的硬核范围。因此，

我们对量化的位置进排序，并从下上调整位置。位置调整先检查可的硬核是否以放置该列剩余的级联链，如果不能则将当前级联链向下调整到所需位置，来使超出范围的硬核放置合法化。另 ，如果当前组与前组重叠，则将其调整为向上移动，直到合法放置为 。 23

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 3 章基于进化算法的 FPGA 脉动阵列快速硬核布局法算法 3.1: Distribution genotype legalization

```
1 hardBlocks = {DSP48, BRAM, URAM};
2对于每个 hardBlock in hardBlocks 进行
// distribution: a float array
3 distribution = GetDistribution(hardBlock, genotype);
4 distribution = Normalize(Distribution);
// quantize
5对于 i ← 0 转到 distribution.length 进行
6 distribution[i] = distribution[i] × blockSum;
7 distribution[i] = Min(distribution[i], columnMax);
// handle rounding error
8 count=0;
9当 count < blockSum - Sum(distribution) 进行
10 col_idx = 0;
11当 col_idx < all_columns 进行
```

12如果 distribution[col_idx] + 1 ≤ column_max 则

13 distribution[col_idx] += 1;

14 count++;

15 col_idx++;

16如果 count == block_sum - Sum(distribution) 则

17 break;

算法 3.2: Location genotype legalization

1 hardBlocks = {DSP48, BRAM, URAM};

2对于每个 hardBlock in hardBlocks 进行

3 location = GetLocation(hardBlock, genotype);

// partition location into columns

4 location_columns = Partition(location, distribution);

5对于每个 l_column in location_columns 进行

6 Sort(l_column, ascending);

7对于 i ← 0 转到 l_column.length 进行

8 l_column[i] ×= columnSize;

9 l_column[i] = Int(l_column[i]);

10 need = (l_column.length-i-1) × group_size;

11 avail = columnSize - (l_column[i] + group_size);

12如果 need > avail 则

13 l_column[i] -= need - avail;

14否则如果 l_column[i] < l_column[i-1] + group_size 则

15 l_column[i] = l_column[i-1] + group_size; 24

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 3 章基于进化算法的 FPGA 脉动阵列快速硬核布局法由于 Mapping 基因型只优化编码的顺序， 不对编码的值进修改，所以不会产不合法编码，也就不需要解合法化。以上内容详细阐述了解合法化的步骤，这些步骤保证了布局结果满三个约束条件，并且与解析布局法 (Analytical Placement) 相 ，

我们的合法化步骤开销更低。

3.3 进化算法：NSGA-II 与 CMA-ES

在建了布局模型的基础上，本节详细阐述本使的进化算法优化法。进化算法于布局问题的优化相传统的模拟退与前泛应的解析法相主要有三点优势：（1）与模拟退相 ，进化算法基于种群的优化式个体之间相互独 ，天然具有规模并化加速的优势；（2）多标遗传算法的发展使得进化算法相其他法更擅长优化多个标函数，在本中我们利这优势设计了两个标函数分别优化线长和计算单元边界框，取得了单标更好的优化效果；（3）基于概率分布的进化算法在同样的问题规模可以在优化速度和结果质量上同时超越 state-of-the-art 的解析布局算法。本节详细介绍 RapidLayout 布局引擎所使的两种进化算法： 配排序遗传算法与协差矩阵适应进化策略。

3.3.1 非支配排序遗传算法：NSGA-II

配排序遗传算法 (Non-Dominated Sorting Genetic Algorithm, NSGA-II[15]) 是

种较早提出的多标遗传算法。近年来由于强化学习[12] 和 NAS(Neural Architecture

Search)[

13]中的应再度流 。图3-3描述了 NSGA-II 的优化过程，其核内容是两个排

序算法：Non-dominated Sorting 与 Crowd Distance Sorting。

Non-dominated Sorting 也称为配排序法，是 NSGA-II 处理多个标的核法，其思想是 “ 配”的概念进排序：如果个体 A 在各个衡量标准（多个标函数值）上均劣于个体 B，则 A 配 B。在种群中可以根据配关系将个体分为许多组，

每组内的个体互相不配，每组内的个体和其他组的个体定存在配关系，这样的组成为帕累托前端（Pareto-Front)。简之，因为多标的存在，在排序过程中只能将个体分为组， 组内的个体法直接判断孰优孰劣。这种基于配标

准在多标维度上对个体进分组并排序的法称为配排序法 (Non-dominated Sorting)。如图3-3所，第轮淘汰使的此种排序法，但如果同前端需要淘汰部分个体，则需要其他法。Non-dominated Sorting 的排序算法请见附录A 算法 A.1。 25

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 3 章基于进化算法的 FPGA 脉动阵列快速硬核布局法

Pt

Qt

F1

F2

F3

Non-dominated

Sorting

Crowding

Distance

Sorting

Rt

Rejected

Pt+1

图 3-3 NSGA-II 的优化过程[15]： 先亲代通过交叉变异产同等数量的代，然后通过 Non- dominated Sorting 将种群个体分为多个帕累托前端（Pareto-Front），并淘汰部分劣势个体。

如果同个帕累托前端内的个体需要淘汰，则通过 Crowd-Distance Sorting 保留具有代表性的

代，最终淘汰半种产下代亲代。

Crowd Distance Sorting 于解决同个帕累托前端内的排序问题。此种排序法的核思想是保留同帕累托前端内部具有“代表性”的个体。其具体实现法是先衡量所有个体在所有标函数维度上与其邻近的个体之间的距离，根据距离先计算出

所有点的 crowd distance 值，其具体做法于附录B算法 B.1中详细说明。然后根据 crowd distance 排序，最终与其邻近个体在所有标函数维度上距离的被淘汰，距离的被保留，也就是保留了具有“代表性”的个体，实际此种排序淘汰算法促使种群个体在所有标函数维度上均匀分布。Crowd Distance Sorting 的排序算法请见附录B 算法 B.2。

3.3.2 协方差矩阵适应进化策略：CMA-ES

协差矩阵适应进化策略 (Covariance Matrix Adapted Evolution Strategy, CMA-ES)

是实数域上的种优化法，适合于箱问题、 线性优化、病态 (ill-conditioned) 系统优化，并且需导数[16]。CMA-ES 是种 state-of-the-art 的进化算法，被于包括神经

络参数优化[14] 在内的多个领域。

CMA-ES 需求导的特性使其适于许多基于梯度优化法失败的问题，或些平滑， 连续域的优化问题。CMA-ES 将产可解的过程建模为均值向量为 μ，协差矩阵为 Cσ的斯随机过程采样。在每次进化过程中新的可解使更新后的均值向量和协差矩阵采样得到，然后根据标函数对可解进排序，最后使前 25%

的可解更新均值向量与协差矩阵。算法3.3详细描述了 CMA-ES 的实现原理和参数设置。 26

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 3 章基于进化算法的 FPGA 脉动阵列快速硬核布局法参数设置：

λ = 4 + 3ln(n) , μ = λ/2 , wi=l n(μ+1) ln(i)∑μ j=1 ln(μ+1) ln(j)

(i = 1, ..., μ), μw=1∑μ i=1 w2i , cσ=μw +2 n+μw+3

, dσ= 1 + 2max(0, √

μw 1 n+1

1 + cσ), c c=4n +4

, μc ov= μ w, c cov= cd ef cov = 1

μcov 2

(n+ √ 2)2

+ (1 1/μc ov)min(1,2

μcov 1

(n+2)2+μcov)

参数初始化： g = 0, B = I, D = I, pσ= (0, .., 0)T, pc= (0, ..., 0)T, C = I, x w∈ (R) n, σ ∈ R

算法 3.3: The (μ/μW, λ) CMA-ES Algorithm

1当 Stop Criterion not Reached 进行

2 g ← g + 1

3 zi～ N (0, I) for i = 1, ..., λ

4 xi= x w+ σBDzi

5 x w=

∑μ i=1 wixi:λ; // x i:λdenotes the i-th best individual out of the λ

6 z w=

∑μ i=1 wizi:λ; // z i:λdenotes the i-th best mutation vector

7 pσ← (1 c

σ)p

σ+ √ cσ(2 c

σ) √

μwB z w

8如果 ‖pσ‖/ √

1 (1 cσ)

2g < (1.4 + 2/(n + 1))E(‖N (0, I)‖) 则

9 Hσ= 1

10否则

11 Hσ= 0

12 pc← (1 cc)p c+ Hσ √ cc(w c c) √

μwBD z w

13 C ← (1 cc ovC) + 1/μ covc covpcp c

T+ cc ov(1 1/μ cov)

∑μ i=1 wiBDzi:λ(BDz i:λ)T

14 σ ← σ exp(cσ/d

σ(‖p

σ]‖/E(‖N (0, I)‖)) 1)

15 B D2= eigendecomposition(C)

为了适应规模优化问题，本修改了协差矩阵的更新法[43]，令 D2 = diag(C), cc ov=n +2 3 cd ef cov

，也就是将协差矩阵的更新约束在对线，其余元素置零，并提了学习率。这种优化法使得 CMA-ES 的空间复杂度由次减低为线性，加快了规模问题的优化速度。

3.4 RapidLayout 布局框架上节详细阐述了布局问题的建模、 标函数、优化约束，以及本提出的进化算法基因型设计、解合法化步骤。在此基础上，我们基于 Xilinx RapidWright FPGA 实现框架设计并且实现了端到端的布局布线框架，RapidLayout。RapidLayout 的核部分是上节中描述的进化算法布局引擎，此结中详细阐述 RapidLayout 的设计作流程，各实现步骤的含义，内容，及实现法。图3-4展了 RapidLayout 的作流。 27

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 3 章基于进化算法的 FPGA 脉动阵列快速硬核布局法

Convolution Block DCP

Netlist Replication [<1s]

A

Evolutionary

Hard Block

Placement

[30 s–5 min]

B

Placement and

Site Routing [≈3 min]

C

Post-Placement

Pipelining [≈10 s]

D

SLR Placement and Routing [≈54 min]

E

SLR Replication [≈2 min]

F

Compute objective

Generate candidates

Update

RapidWright

Vivado

Bitstream ev o lv e

图 3-4 RapidLayout 作流，以 Xilinx VU11P FPGA 上的实现流程为例。主要的布局优化和物理实现作由 Rapi dWright API 开发完成，Vivado 只负责控制逻辑布局以及 SLR 内的布线。 28

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 3 章基于进化算法的 FPGA 脉动阵列快速硬核布局法

A 逻辑网表复制： RapidLayout 的输是个卷积计算单元的逻辑表，以个DCP checkpoint 件的形式存储。Rapi dLayout 先根据选定的 FPGA 器件型号和卷积计算单元内的各硬核例计算出最小可复制矩形域（Minimum Replica ting Rectangle）

同时确定矩形域最可放置的卷积单元个数。最可复制矩形域是进布局案优化的区域，其具体含义为：根据设计使的硬核例与物理硬核资源例确定的最矩

形，在此矩形内可使各硬核利率达到最 ，同时其布局结果可于复制填充整个FPGA。确定布局范围以及该范围内卷积计算单元个数之后，RapidLayout 复制输的逻辑表并连接全局信号 (reset, clock 等)。以 VU11P FPGA 为例，RapidLayout 确定的最可复制矩形范围是宽为 8 个时钟域， 为 2 个时钟域，可放置 80 个卷积运算单元，

三种硬核的利率均在 95% 以上。这最可复制矩形区域内的布局结果可经过 2 次

复制充满个超逻辑域，可经过 6 次复制充满整个 FPGA。

B 基于进化算法硬核布局：RapidLayout 使配排序遗传算法 (Non-Dominated

Sorting Genetic Algorithm II, NSGA-II) 或协差矩阵适应进化策略 (Covariance Matrix

Adapted Evolution Strategy, CMA-ES) 两种进化算法对硬核布局优化。进化算法包括产可解，计算适应度，更新可解等操作，在计算适应度的步骤需要根据布局法

计算线长。由于进化算法是种重复优化法 (Iterative Method)，所以我们不可能依赖

Vivado 估计每个可解的线长，否则会需要依次进物理布局，导致优化时间过长。

RapidWright 的优势在于可以快速获得 FPGA 的物理信息，因此本利 Rapidwright 快

速计算出板上物理位置之间布线所需的线长。

C 物理网表布局和 Site 布线： 优化过程结束，RapidLayout 会得到最可复制矩形内的最优布局结果。然后根据这结果将逻辑表和布局结果复制到整个超逻辑域（SLR），再修改物理表，进实际的物理布局操作。本使 Rapid Wright 提供的 FPGA 物理层 API 开发了三种硬核的布局法， 先将硬核放置在对应的物理位置

（Site），然后进 “Site 布线”，连接硬核与 Site 内部的信号接 ，这步骤保证了布局之

后导出的 DCP checkpoint 件与 Vivado 的兼容性。

D 流水线： 布局完成之后，硬核的物理位置已经确定，因此可以估计硬核之间的布线线长和需要插的寄存器个数，对较长的数据通路进流线，以得到较的时钟频率。这个步骤在布局之后进 ，是为了保证正确的数据通路插合适的寄存器个数。进流线操作的的是为了让整个加速器设计的时钟频率达到 650MHz 以上。

650MHz 是脉动阵列中 Ultra RAM 运的上限，通过流线插寄存器的式我们可以

获得于 650MHz 的时钟频率，但实际运中的时钟还是需要设置在 650MHz。此步骤在 URAM → BRAM 和 BRAM → DSP 之间的数据通路插流线寄存器，数根据线 29

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 3 章基于进化算法的 FPGA 脉动阵列快速硬核布局法

长确定。

E 超逻辑域布局与布线： RapidLayout 负责完成硬核布局布线，控制逻辑和流线寄存器等使的 LUT 和 Flip-Flop 则又 Vivado 完成布局和布线。在此步骤中，流线之后的设计先被导出成 DCP checkpoint 件，然后由 Vivado 读完成超逻辑域的

布局与布线，完成布局布线的设计也以 DCP checkpoint 的形式存储。

F 超逻辑域复制：在上步步骤中 RapidLayout 完成了单个超逻辑域的完整实现，

以 VU11P 为例整个 FPGA 中有 3 个超逻辑域，每个超逻辑域内部的器件构造与排布相同。因此，我们使 RapidWright 的硬件 API 开发了 SLR 复制的功能，将个 SLR 的布局布线结果连同逻辑表复制到整个 FPGA 的所有超逻辑域，由此获得了幅的时间节省。

对于 VU11P FPGA 来说，RapidLayout 端到端的作流与 Vivado 的实现作流相要快 ≈5–6×。RapidLayout 的整个作流仅需约 1 时，并且需输布局信息，

整个流程动化；Vivado 则需要以 XDC Constraints 的式输完成的硬核布局，

否则其布局引擎法产可布线的结果，并且整个作流耗费 5–6 时。 的硬核布局需要经验和长达周的反复调试才能使时钟频率达到 650MHz 以上， RapidLayout

使进化算法布局引擎可以在分钟的时间内动发现质量的硬核布局结果， 需动调试和先前经验。

3.5 RapidLayout 以 Xilinx VU11P FPGA 为例的布局实现过程

上节介绍了 RapidLayout 布局框架的作流，本节以 Xilinx UltraScale+ VU11P

FPGA 的实现为例阐述布局的三个过程，从单个卷积运算单元的布局到整个 FPGA 的完全实现。

单个卷积运算单元的布局： 图3-5展了单个卷积运算单元的布局结果。本为硬核布局设计了套可视化套件，包含在 RapidLayout 框架中，包含单个卷积单元的位置 0 100 200 300 400 500 600 0 50 100 150 200 250 300

DSP48

BRAM

URAM

图 3-5 个卷积计算单元的布局结果可视化。此卷积计算单元的 RTL 设计如图 reffig:sys 所 。其中布局使的硬核亮标出，整体的布线范围使灰阴影区域标出 30

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 3 章基于进化算法的 FPGA 脉动阵列快速硬核布局法 0 500 1000 1500 2000 25000 250 500 750 1000

图 3-6 最可复制矩形区域内的布局结果可视化。此最可复制矩形域放置了 80 个卷积计算单元，

图3-5的位置在图中标出作例参考信息、整体的布局可视化结果、以及布局过程的动态展 。图中浅矩形为三种硬核的分布列，其中被选中的硬核位置被亮标出，整个卷积核的分布范围使灰阴影标出。从图中可以发现，URAM、BRAM、DSP 资源列分布不规则，这导致了单个卷

积运算单元的布局不规则，也说明了单个卷积单元的布局结果法直接通过复制-粘贴

(copy-paste) 的式适于所有卷积单元。

一个最小可复制矩形区域：RapidLayout 根据器件型号和设计中使硬核的例动确定最可复制矩形区域，最化每种硬核的使率来得到最的计算密度，同时便于布局结果复 ，最化优化问题规模从加快速度。在图3-6中展了 VU11P FPGA

最可复制矩形区域内的布局结果，使与图3-5相同的可视化法。该最可复制矩

形区域宽 8 个时钟域、 2 个时钟域，放置了 80 个卷积运算单元，URAM 利率 100%，

DSP48 利率 93.7%，BRAM 利率 95.2%，完整 FPGA 实现的硬核利率与此相同。

4. 基于进化算法的FPGA脉动阵列快速硬核布局方法_第4部分

总字数：**11425**

相似文献列表

去除本人已发表文献复制比：1%(113) **文字复制比：**1%(113) **疑似剽窃观点（0）**

1	基于L-模糊集的P2P信任模型及应用研究 权义宁(导师：胡予濮) -《西安电子科技大学博士论文》- 2009-12-01	0.5%（54） 是否引证：否
2	基于进化优化的类集成测试序列生成方法研究 张悦宁(导师：姜淑娟) -《中国矿业大学硕士论文》- 2019-05-01	0.5%（54） 是否引证：否
3	硅锗团簇的结构、稳定性及电子性质的理论研究 秦薇(导师：吕文彩) -《吉林大学博士论文》- 2010-06-01	0.5%（52） 是否引证：否
4	R企业拖车配送路径优化问题研究 汤浩晨(导师：胡小平;邢晨) -《东南大学硕士论文》- 2018-06-07	0.5%（52） 是否引证：否
5	回归测试的测试用例优先级排序问题研究 赵晓彬(导师：王寒) -《天津大学硕士论文》- 2015-11-01	0.4%（51） 是否引证：否

原文内容

从图中的灰重叠区域将可以观察到阻塞 (Congestion) 情况，RapidLayout 的布局引擎可以最化重叠，从减布线资源的阻塞。

完整 FPGA 实现： 图3-7展了 RapidLayout 从最可复制矩形区域的布线出发完成整个 FPGA 布线的步骤。 先 RapidLayout 将最区域内的布线结果复 ， 复制粘贴直到布满个超逻辑域，在 VU11P 上则是复制 2 次充满了 SLR 0。随后，RapidLayout

调 Vivado 完成控制逻辑的布局与完整布线，得到了个超逻辑域（SLR0）的完整实现。下步，RapidLayout 将个超逻辑域（SLR0）内部的完整实现通过复制粘贴的式复到其他超逻辑域 (SLR1, SLR2) 上，得到完整的 FPGA 实现，可产特流。 31

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 3 章基于进化算法的 FPGA 脉动阵列快速硬核布局法

SLR0R epeat Rect. (Fig 3-6)

SLR1

SLR2

C o p y

P lacem en ts +

V iv ad o

P +

CR o p y

P lacem en t +

R o u tin g in

R ap id

W rig h t

F

C

D

E

图 3-7 从最可重复矩形区域内的布局结果到完整 FPGA 实现过程意图。 先，最可重复矩形内的布局结果经过次复制充满 SLR0，RapidLayout 对布局结果和逻辑表同时进复

制。然后 RapidLayout 调 Vivado 完成 SLR0 的布局布线，得到 SLR0 的完整实现。最后，

RapidLayout 使 RapidWright API 将 SLR0 的布局结果和布线结果复到 SLR1 和 SLR2，得到完整的 FPGA 实现。

3.6 本章小结本章先介绍了硬核布局建模为多标优化问题的法，明确了问题的优化标

与约束，阐述了本的创新基因型设计与相应的解合法化过程，明确 NSGA-II 与 CMA-

ES 的算法原理与实现案，最终介绍了 RapidLayout 布局框架的作流，并以层层递进的式举例阐述了脉动阵列的布局实现过程。 32

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 4 章实验结果与分析

第 4 章实验结果与分析

前章节详细介绍了本的进化算法硬核布局的原理与实现式，并介绍了 Rapid-

Layout FPGA 实现框架的原理与作流程。本章通过对实验设计研究进化算法布局与模拟退 、解析布局等法性能的较，并验证迁移学习的有效性。 先，本章介绍实验中涉及的 FPGA 器件与对算法，然后介绍实验的编程实现、运环境、参数确定等实验细节。本章设计了多种可视化法，展本提出的进化布局算法与 state-of-the-art

法的性能、收敛过程对 ，并详细讨论了流线级数对时钟频率的影响。最后，本章评估 RapidLayout 的迁移学习性能并展布局结果。

4.1 Xilinx UltraScale+ FPGA 介绍表 4-1 实验器件 Xilinx UltraScale+ Virtex 资源列表

型号 SLR 数 DSP 数 BRAM 数 URAM 数 Flip-Flops(K) LUTs(K)

VU3P 1 2280 1440 320 788 394

VU5P 2 4560 2880 640 1201 601

VU7P 2 4560 2880 640 1576 788

VU9P 3 6840 4320 960 2364 1182

VU11P 3 9216 4032 960 2592 1296

VU13P 4 12288 5376 1280 3456 1728

表4-1列出了本实验中使的 FPGA 及其计算资源。UltraScale+ 是 Xilinx 推出的端 14/16nm 系列 FPGA，搭载了 DSP48，Block RAM 和 Ultra RAM 三种计算存储硬核资源。RapidLayout FPGA 布局框架是向 UltraScale+ 系列 FPGA 设计的， 持除了 HBM（High-Bandwidth Memory）型号以外的全部 FPGA。

4.2 对比布局算法本选取了多个布局算法作为对实验，包括：（1）模拟退算法；（2）基于模拟退的标准 FPGA 实现具 Versatile Place and Route[44]；(3) State-of-the-art 解析布局算法 UTPlaceF[8]；（4）单标遗传算法；（5）RapidWright 内部布局引擎（基于退 [30]；

（6）Vivado 解析布局引擎。本节阐述以上对实验的原理简介与实现法。 33

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 4 章实验结果与分析

4.2.1 模拟退火算法 Simulated Annealing

模拟退算法 (Simulated Annealing) 是种基于概率的启发式搜索优化算法，它将系统模拟为温熔化材料逐渐凝固的过程，每轮迭代对当前系统状态做出随机变化，

通过温度函数控制接受该随机改变的概率，经常被在离散系统的优化，FPGA 布局问题是其经典应之 。退算法已经被证明，对于 FPGA 布局问题在限长的时间内可以保证收敛到全局最优值。但是，退算法的收敛速度和结果质量极受到冷却进度表 (Cooling Schedule) 的影响：冷却速度过快会导致系统收敛到局部最优点，速度过慢则会导致收敛时间显著变长。 般来说，冷却进度表的确定是与优化问题相关的，也

就是需要针对具体优化问题调节参数。本中使 Opt4J 优化框架的退优化引擎，

Java 实现了 FPGA 硬核布局的模拟退搜索法。第4.4.1节对于冷却进度表的确定进实验并给出了最优结果。由于退算法仅持单标优化，本将提出的两个标函数乘积作为退算法的优化标函数。

4.2.2 Versatile Place and Route

Versatile Place and Route (VPR) 是多伦多学提出的 Verilog-to-Route(VTR)[44] FPGA

实现框架的布局布线具，是学术界最常的标准对基线。VPR 基于模拟退算法，

提出了种线性阻塞 (linear congestion) 标函数改进，其形式如式4.1确定。

Cost =

Nnets∑ n=1 q(n)[bbx(n)

Ca v,x(n) + bby(n)

Ca v,y(n)] (4.1)

其中求和是对于所有的信号线，bbx和 bb y

代表每个信号线边界框的宽和 ， q(n)

是对实际线长估计的补偿值，Ca v,x

和 Ca v,y

是横纵向平均的布线通道容量。本采了最新的 VPR 7.0 Official Release 进制件作为对 。由于 VPR 使的架构与表

格式与 RapidWright 不同，本基于 Python 开发了以下具：（1）将 Xilinx UltraScale+

VU11P 架构转化为 architecture.xml 架构信息转化具；（2）将本使的脉动阵列表转化为 Berkeley Logic Interchange Format 的具， 成 conv_model.blif 件；

（3）将 VPR 布局结果转化为 RapidWright 和 Vivado 持的 XDC 格式的转换具。以上具链均与 RapidLayout 集成，在对实验时由 RapidLayout 动调 。 34

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 4 章实验结果与分析

4.2.3 UTPlaceF 解析算法布局工具UTPlaceF 是前最先进的解析布局算法具，由德克萨斯学奥斯汀分校[8] 提出，

并于 ISPD2016 Routing-Aware Placement Contest 获得等奖。UTPlaceF 使 HPWL(Half-

Parameter Wirelength) 作为标函数，其形式为：

HP W L(x, y) = ∑ e∈E maxi

,j∈e|xi xj| + maxi,j

∈e|yi yj| (4.2)

其中 E 为全部信号线，x, y 为信号线连接的全部计算单元 (cell)。UTPlaceF 使欧距离平对 HPWL 近似从使其可导，然后使 Quadratic Programming 求解布局结果。本使 UTPlaceF TCAD Version 公开的进制件进对实验。由于 UTPlaceF

使 Extended BookShelf Format 作为架构和逻辑表的定义格式，本同样开发了表和布局结果的转换具，与 RapidLayout 集成， 动调 。

架构近似：由于 UTPlaceF 仅向 UltraScale 架构[8]，其架构定义不包括 Ultra RAM

硬核。因此，本根据硬核做出如下近似：UltraScale+ RAMB18 → UltraScale DSP48,

UltraScale+ URAM288 → UltraScale RAMB36，并保持硬核数量、列分布与 UltraScale+

VU11P FPGA 相同。

4.2.4 遗传算法 Genetic Algorithm

遗传算法 (Genetic Algorithm, GA) 是最基本的进化算法。本采 Opt4J[42] Java

优化框架实现了单标的 GA 硬核布局引擎。遗传算法采最基本的交叉 (Cross-over)

和变异 (Mutation) 产新的可解，但仅持单标函数。本采与 NSGA-II 相

同的问题建模， 标函数为卷积计算单元最边界框尺 ，基因型设计与参数设置和

NSGA-II 保持相同。本实验的的在于说明多标优化对布局结果质量提升的影响。

4.2.5 RapidWright 布局引擎RapidWright 持基于 Modular Design[30] 的实现式，可以将个卷积运算单元包装为个 Module，使内置的退引擎进布局。但实际试验结果如图4-1所， 法在 VU11P FPGA 上完成对 480 个卷积运算单元的布局。失败的原因在于：RapidWright

先对 Module 分析，得到整个 FPGA 上所有的可布局位置。但由于硬核的列分布不规律，RapidWright 仅能发现同列的可布局位置。在图4-1中，RapidWright 仅发现了 12 个可布局位置，因此其退引擎法给出对于 480 个卷积计算单元的合法布局结果。 35

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 4 章实验结果与分析图 4-1 RapidWright 退布局引擎在 VU11P FPGA 上的布局结果

4.2.6 Vivado 解析布局引擎Vivado 的布局引擎是多标的解析布局引擎。但对于本中的 480 个卷积计算单元法在不通过布局的情况下给出可布线的布局结果。

图4-7 (a) 展了 Vivado 在不给出布局的情况下仅依赖内置的解析布局引擎得到的尝试布线结果，图中的红连线表布线失败的信号。由于 Vivado 给出的布局结果质量差，布线阶段纵向布线资源耗尽所以 FPGA 实现失败。

图4-7 (b) 展了 Vivado 在给出全部硬核布局的实现结果。由于 Vivado 法通过解析布局引擎完成设计布局，所以需要通过设计的式将 480 个卷积

运算单元的所有硬核布局位置确定，并通过 XDC Constraint 的式输 Vivado，

Vivado 仅完成控制逻辑的布局。 设计的式通常需要消耗周的时间试错才能最终得到可以达到 650 MHz 时钟频率的结果，所需的作量巨 。尽管如此，在给出硬核布局的情况下 Vivado 的完整实现仍需 5–6 时才可完成。与此相对的是，RapidLayout 可以动发现硬核布局，保证达到 650 MHz 以上的时钟频 36

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 4 章实验结果与分析

率，不但避免了量的作， 且整体实现时间缩短到 1 时左右。

Vivado 的布局实验再次说明了硬核利率的设计和不规则的硬核与分布，

为布局布线带来了很的难度。即使是 Vivado 这样成熟的商业化解析布局引擎也法动完成布局。因此，开发 RapidLayout 基于进化算法的多标快速布局引擎是有必要的。

4.3 实验的编程实现

4.3.1 运行环境

表 4-2 实验测试环境CPU Intel Xeon Gold 6230 40-Thread

操作系统 Ubuntu 16.04 LTS

内存 128 GB

4.3.2 软件工具

表 4-3 实验软件具

RapidWright 2019.12 Beta Pre-release

Vivado Design Suite 2018.3

Opt4J[

42]3.1.4 2015-release

Apache Commons Math Optimization Library[?]3.4

表4-3列出了本实验使的软件具。RapidLayout 基于 RapidWright Java 和Python 开发，其中 NSGA-II, 模拟退，单标遗传算法布局引擎使 Java 的 Opt4J

优化库实现。对规模优化问题改进的 CMA-ES 布局算法通过 Apache Commons Math

Optimization Library Java 数学库实现。辅助性套件中，第3.5节中展的可视化具由Python Matplotlib 实现，针对 UTPlaceF, VPR 对实验的表、架构转换具由 Python

实现，布局结果转换具由 Java 实现。以上各个套件集成在 RapidLayout 具链之中，

通过配置件控制 RapidLayout 的作流和输出，组成了完整的端到端 FPGA 实现框架。 37

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 4 章实验结果与分析

4.4 实现细节与敏感度分析本节详细阐述模拟退 ，CMA-ES，NSGA-II 算法的参数选取与敏感度分析，通过实验验证其合理性并得出结论。

4.4.1 模拟退火参数确定通过第4.2.1节对模拟退算法的分析可知，冷却进度表很程度影响退的收敛速度与结果质量。因此，本节通过实验来验证不同冷却进度表对收敛速度与结果的影响程度，并选取最优参数进与进化算法的对实验。

0 200 k 400 k 600 k steps 101 8 101 9 102 0

B b o x

S ize × w ire le n g th2 exponential hyperbolic linear

图 4-2 模拟退布局算法在指数、双曲、线性三种类型冷却进度表条件下的标函数收敛进程图4-2展了模拟退算法分别在指数、双曲、线性三种类型的冷却进度表下标函数的收敛过程。实验过程中，三种冷却进度表分别选了组参数组合，并使相同的 early-stopping 标准。从图4-2观察可以得知：

冷却进度表类别的不同对收敛的影响于其内部参数的变化。

线性冷却进度表收敛结束最快，但停在了局部最值。

指数冷却进度表能够达到近似全局最优，但所花时间过长。

双曲线冷却进度表表现最优，所花时间少与指数函数能够达到近似全局最优，

因此选择双曲线冷却进度表作为模拟退布局引擎的冷却进度，并选择取得收敛后标函数值最的组参数作为默认参数。 38

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 4 章实验结果与分析

4.4.2 CMA-ES 敏感度分析 sigma 0.20.40 .60.81 .01.21 .4 population2550751001251 50 175 wirelength 400 0 5000 6000 7000 8000 9000 10000

图 4-3 CMA-ES 敏感度分析，改变 sigma 与 population 对收敛线长的影响CMA-ES 有两个主要的可选参数，个是 σ 初始步长，另个是采样数 population。

σ 对 CMA-ES 的优化过程影响主要有两点：（1）如果 σ 设置过 ，则容易陷局部最值；(2) 如果 σ 设置过 ，则容易造成结果不收敛。实验中采取了 $\sigma \in [0.1 : 0.1 : 1.5]$

和 $population \in [20 : 10 : 180]$ 的参数组合范围，每组参数运 CMA-ES 布局算法 10

次，取最值作为该组合的最优表现。得到实验结果如图4-3 所 ，可以观察到 $\sigma > 1.2$

会导致结果难以收敛， $\sigma < 0.4$ 则会收敛局部最值，符合对 CMA-ES 的分析。 population 的改变对优化效果影响不 ，因此选择较的 population = 30 作为默认参数，以加快 CMA-ES 的优化进程。

4.4.3 NSGA-II Reduced

图3-2介绍了本的“三段”结构基因型设计，分别是 Distribution, Location, 和 Map- ping。Distribution 指定了布局任务选的硬核在列中的分布，确定了每列选择硬核的个数；Location 在此基础上确定每个硬核在列中的位置，从完全确定被选中硬核的物理位置；Mapping 在选中位置的基础上规定逻辑硬核与物理位置的对应关系，从完全确定个布局可解。这种“三段”结构的基因型可以充分表达问题的解空间，为搜索到全局最优解提供了够的由度。

本节提出种 trade-off 的思路，取消三个基因型中的 Distribution 和 Location, 按照均匀分布的式将选取硬核的数分配到各列，再从下上依次以“堆叠”的式选取 39

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 4 章实验结果与分析硬核位置。这样，仅优化逻辑硬核与物理位置的对应关系，意味着放弃了部分解空

间，减了搜索的由度，来加速搜索过程。本将这种仅优化 Mapping 部分基因型的NSGA-II 算法称为 NSGA-II Reduced, 并采与 NSGA-II 相同的参数设置进性能对实验。

4.5 实验设置以及实验结果本章以上节详细介绍了对实验的原理与最优参数选取法，本节在此内容的基础上进所有算法的布局性能对实验，性能指标包括：运时间、 标函数、收敛稳定程度、布线后设计的时钟频率，以及达到 URAM 上限 650 MHz 时钟频率所需的流线寄存器开销。最后，本节展 RapidLayout 的迁移学习性能，即布局结果迁移到不同型号 FPGA 的能 。

4.5.1 实验设置表 4-4 实验参数设置

布局算法参数设置

NSGA-II population=50, mutation=0.99

NSGA-II(Red) population=50, mutation=0.99

CMA-ES σ = 0.5, population=30

SA 双曲冷却进程表: t = t0(tnt0)i/n, t0= 1000, tn= 100

GA population=50, mutation=0.99

VPR 使进制件，默认参数

UTPlaceF 使进制件，默认参数Manual 参数表4-4中列出了布局性能对实验所有法的参数设置。由于 RapidWright 法产布局结果，所以不纳性能对实验；Manual 表布局结果，也就是设计

480 个卷积计算单元硬核在 VU11P FPGA 的布局结果，以 XDC Constraints 的形式输

Vivado 进对实验。另外需要说明的是，由于 UTPlaceF 和 VPR 布局实验法设置

式3.5中提出的级联约束条件，因此得到的布局结果不满 Xilinx UltraScale+ 的架构布局布线要求，所以法完成时钟频率和流线开销实验，但仍可以衡量其布局结果的标函数值。所有对实验均在同实验软件和硬件条件下完成。40

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 4 章实验结果与分析

4.5.2 性能对比本节对所有优化算法采了设置随机种的初始化，每种优化算法运 50 次，随机选取其中的 10 次结果进完整实现（布局、布线、Vivado 报告时钟频率）对时钟频率结果。 0.2 0.4 0.6 0.8 1.0 1.2 wirelength2

1e8 0 500 1000 1500 2000 2500 runtime (sec) 1000 2000 3000 4000 5000

BboxSize 0 500 1000 1500 2000 2500 0 1 2 3 4 wirelength2× BboxSize

1e11 0 500 1000 1500 2000 2500 runtime (sec) 500 550 600 650 700 750 clock frequency (MHz) 0 500 100 0 1500 2000 2500

Annealing

NSGA-II

NSGA-II(Red)

CMA-ES

GA

VPR

UTPlaceF

图 4-4 退算法、NSGA-II、NSGA-II Reduced、CMA-ES、GA、UTPlaceF、VPR 的布局结果对：

标函数值与时钟频率图4-4展了所有动布局法的运时间，wirelength2, 最边界框尺， 者相乘 wirelength2 × Bbo xSize，以及 Vivado timing report 的时钟频率对 。表4-5列出了图中内容的详细信息，以及进化算法相对对实验各性能提的倍数。由性能对的

结果可以清晰得出以下点结论：

NSGA-II 退算法快约 2.7 倍，同时可以得到 14.7% 的最边界框尺，但是牺牲了约 12.9% 的线长。然，在时钟频率项 NSGA-II 仍然取得了优于退

的表现，这是由于相退的结果 NSGA-II 的优化结果更加稳定。

CMA-ES 约退快 30 倍。尽管 CMA-ES 得到的最边界框尺和线长相退的结果要差 (≈ 41%, ≈ 16%)，但 CMA-E S 仍然取得了与退相当甚更 41

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 4 章实验结果与分析表 4-5 平均运时间，平均线长，平均最边界框尺，达到 650MHz 最少所需流线寄存器数，

以及平均时钟频率对 。左侧为本提出的 NSGA-II, NSGA-II Reduced, CMA-ES，右侧为对算法。与各个对算法相每个指标的提升倍数在括号内标出，其中红 →NSGA-II，绿

→CMA-ES

Method NSGA-II CMA-ES SA GA VPR UTPlaceF Manual

Runtime (secs) 586 (323) 51 1577 (2.7×, 30.8×) 850 (1.5×, 16.7×) 76 (0.13×, 1.5×) 473 (0.8×, 9.3×) 1–2 wks

Wirelength 3.5K (3.5K) 4.4K 3.1K (0.9×, 0.7×) 9.2K (2.6×, 2.1×) 8.5K (2.4×, 1.9×) 7.8K (2.2×, 1.8×) 8.1K (2.3 ×, 1.8×)

BBox 1183 (1543) 1606 1387 (1.2×, 0.9×) 1908 (1.6×, 1.2×) 4941 (4.1×, 3.1×) 3218 (2.7×, 2.0×) 1785 (1.5×, 1.1×)

Pipeline Reg. 256K (273K) 273K 273K (1.1×, 1×) 323K (1.3×, 1.2×) - - 306K (1.2×, 1.1×)

Frequency (MHz) 733 (688) 708 711 (0.97×, 1.0×) 585 (0.80×, 0.83×) - - 693 (0.95×, 0.98×)

的平均时钟频率 (711 MHz)。

NSGA-II Reduced 这精简后的 NSGA 算法取得了相退快 5 倍的运速度和相完整 NSGA-II 快约 2 倍的速度，但仅仅损失了 2.8% 的时钟频率 (688 MHz)

且仍然于 URAM 限制的 650MHz 最运频率。

与 UTPlaceF 和 VPR 相，NSGA-II 和 CMA-ES 两种进化算法在运时间与结果质量上都有明显提 。

与单标的遗传算法相，NSGA-II 多标优化不但显著提升了结果质量， 且

降低了运时间。这说明本的多标优化设计可以加快优化过程，更有效地使

算法收敛到近似全局最优解。

从表4-5中的结果可以总结出，在 50 次实验中，NSGA-II 和 CMA-ES 相 VPR 和UTPlaceF 得到了更优的结果质量（QoR），相于退来说加速 3–30 倍，仅牺牲可以忽略的结果质量。对于实现时钟频率的标 650 MHz，NSGA -II 退减少 17K（约 6%）

的流寄存器，同时相布局节省 50K（约 16%）的流寄存器。

4.5.3 收敛情况对比

0 10 k 20 k 30 k iterations 107 108 w ire le n g th2

0 10 k 20 k 30 k iterations 2 × 103 3 × 103 4 × 103 max bbox size

0 10 k 20 k 30 k iterations 101 1 101 2

B b o x

S ize × w ire le n g th2

SA NSGA-II NSGA-II(Red) CMA-ES GA

图 4-5 退算法，NSGA-II，NSGA-II Reduced，CMA-ES，遗传算法的收敛进程对图4-5画出了五种算法的收敛过程，包括两个标函数和乘积的优化过程。可以观察到： 42

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 4 章实验结果与分析退收敛所需要的迭代次数最多，因此在 30k 的迭代次数以内标函数降低幅度不明显；

单标遗传算法的优化过程分不稳定，尽管收敛较快但结果质量不；

收敛所需要的迭代次数最少的，也是最稳定的算法是 CMA-ES，在不到 2000 个迭

代次数内已完成收敛，得到的结果质量较；

NSGA-II 和 NSGA-II Reduced 在 15K 个迭代次数之后布局结果质量超过 CMA-

ES。可以观察到去掉 2 个基因型对 NSGA-II 的收敛速度影响不，由此可以判断 NSGA-II Reduced 得到的加速来源于省略的解合法化过程。

4.5.4 流水线级数对时钟频率的影响 0 1 2 3 4 pipelining depth 550 600 650 700 750 clock frequency (MHz)

CMA-ES

NSGA-II

Annealing

GA

Manual

NSGA-II Reduced

图 4-6 六种布局法时钟频率随流线级数的变化图4-6画出了次优化实验中 CMA-ES，NSGA-II，NSGA-II Reduced，遗传算法，退算法以及动布局的时钟频率相对于流线级数的变化。

5. 基于进化算法的FPGA脉动阵列快速硬核布局方法_第5部分

总字数：**6390**

相似文献列表

去除本人已发表文献复制比：5.2%(330) **文字复制比：**5.2%(330) **疑似剽窃观点（0）**

1	基于进化算法的多目标优化方法研究 宋伟(导师：覃俊) - 《中南民族大学硕士论文》- 2011-05-03	4.3%（276） 是否引证：否
2	2009305200309-李煜 李煜 - 《大学生论文联合比对库》- 2013-05-29	0.5%（33） 是否引证：否
3	41100617_孙博_机械工程及自动化_基于ADAMS的万米钻机猫道系统动力学仿真分析 孙博 - 《大学生论文联合比对库》- 2014-06-06	0.5%（33） 是否引证：否

原文内容

NSGA-II 在不需要额外流线的情况下达到了 650 MHz 的实际运频率上限， 退算法，CMA-ES 和动布局少需要 1 级流线。对于规模越的设计，节省级流线带来的寄存器减少就越 。

本在 VU11P 的 480 个卷积单元实验中，NSGA-II 节省的级流线来了 17K–50K 个流线寄存器的减少，极减轻了布线压 。

同时观察到 NSGA-II，CMA-ES 仅需两级流线即可达到 750 MHz 的极时钟频率， 退则需要 3–4 级流线， 布局更是法达到这样的频率。这启着在未来的架构改进和设计改进中如果设计持 750 MHz 以及以上的频率运，NSGA-II

和 CMA-ES 更加有潜 。 43

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 4 章实验结果与分析

4.5.5 布局的迁移学习

表 4-6 硬核布局在不同 FPGA 的迁移学习性能: VU3P, VU11P 作为初始设备

Device

Design Size Impl.Runtime Frequency Placement Runtime

(conv units) (mins.) (MHz) Scratch (s) Transfer (s) xcvu3p 123 46.4 718.9 428.3 - xcvu5p 246 56.9 677.9 396.0 55.7 (7.1×) xcvu7p 246 55.1 670.2 345.4 44.2 (7.8×) xcvu9p 369 58.4 684.9 316.5 45.5 (7×) xcvu11p 480 65.2 655.3 695.9 - xcvu13p 640 69.4 653.2 704.9 58.8 (12×)

RapidLayout 能够在不同资源例、不同硬核列分布、不同的 FPGA 设备上得到质量的布局结果，对布局结果在不同设备之间的迁移学习更加快了布局的搜索进程。RapidLayout 的迁移学习功能将已有设备上的硬核布局结果作为在新设备上布局的初始化参考，从不同例和列分布的新设备上达到随机初始化更快的收敛结果。本将 UltraScale+ 系列 FPGA 根据硬核列数量分为两组，如表4-6所 。分组的依据是硬核资源列的数需要相等或近似，因为初始化将布局结果转化为基因型时，基因型的与硬核资源的列数有关。在这两组中，我们分别令 VU3P 和 VU11P 作为初始设备进布局优化，之后通过迁移学习将布局结果迁移到其他设备。从表4-6中可以总结出，

使迁移学习可以为布局优化过程加速 7–12 倍。

RapidLayout 的 SLR 复制功能使得型 FPGA 的布局布线时间不随资源数量增线性增 。这是因为在 RapidLayout 的作流中，最耗时的部分是单个 SLR 内的布线过程， 旦个 SLR 布线完成，则可以直接将该结果于整个 FPGA，所以 FPGA

的尺增不会导致运时间相应增 。从表4-6的结果可以发现，从 VU3P 到 VU13P，

FPGA 可容纳卷积计算单元的数由 123 增到 640，也就是增加了 5.2 倍； 其实现时间仅有 46 分钟增加到了 69 分钟，仅增了 1.5 倍。 44

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 4 章实验结果与分析

4.5.6 布局结果展示

(a) Vivado 内置解析布局引擎结果，

因阻塞严重法完成布线

(b) 由 Vivado 实现的布局结果，

耗时 5–6 时

(c) RapidLayout 的动布局布线，耗时 ≈1 时图 4-7 480 个卷积运算单元在 Xilinx UltraScale+ VU11P FPGA 上的布局结果

4.6 本章小结本章先介绍实验中涉及的 UltraScale+ VU3P-13P FPGA 器件，使的对算法，

然后介绍实验的编程实现、运环境、参数确定、敏感性分析等实验细节。其次，本章设计了多种可视化法展与 state-of-the-art 法的性能、收敛过程对 。验证了多标进化算法在布局任务上相对其他法的优势：相对退算法 3–30 倍的搜索时间加速，

最先进解析布局法更的边界框尺与线长，相布局节省 6%-16% 的流寄存器等。本章详细讨论了流线级数对时钟频率的影响，从结果中体现出进化算法的另个优势：仅需较少的流线级数即可达到 700 MHz 以上的超时钟频率。最后，

本章评估 RapidLayout 的迁移学习性能并展布局结果，体现了基于 SLR 布局布线结果复的式对规模设计的速度优势。 45

基于进化算法的 FPGA 脉动阵列快速硬核布局法第 5 章总结与展望

第 5 章总结与展望

本章先进了本作的总结，包括本的创新点以及研究意义。然后分析了本提出算法与框架的不之处，最后根据当前学术领域的发展趋势展望未来作。

5.1 工作总结随着深度神经络的发展以及后模摩尔时代的到来，芯领域的研究兴趣逐渐从计算密度的提向计算架构的创新转变。因此，领域定制架构（DSA）和基于 FPGA 的计算架构设计得到了充分发展。 在神经络加速器的规模逐渐增 ，计算速度不断提的过程中，EDA 领域的问题和挑战重新回到了研究者的视线之内。以规模脉动阵列的 FPGA 布局为例，由于其较的资源利率与前异构 FPGA 结构的不规则性，

法使现有的布局具得到可布线的结果，只能依靠设计。针对这问题本提出 RapidLayout 框架，基于进化算法解决布局难题，并加速 FPGA 设计进程。

全的主要作以及贡献点如下：

1) 本将异构 FPGA 的硬核布局建模为多标优化问题，应 NSGA-II 与 CMA-ES

两种进化算法求解，在运时间和结果质量的多个超越 state-of-the-art 布局法。

2) 本针对脉动阵列的级联规则造成的布局限制，提出种创新的基因型设计，将

布局问题划分为三个问题并将布局限制编码进优化过程；

3) 本设计和实现了动化、端到端的快速 FPGA 布局布线框架，RapidLayout，

加速了 FPGA 的设计进程；

4) 本提出了布局的迁移学习法，显著加速了硬核布局结果跨设备优化过程。

5.2 研究展望本的布局框架虽在 UltraScale+ 全系列 FPGA 上验证了通性，但对其他类型加速器的持尚待开发，因此可从以下向进优化：

1) 增加 RapidLayout 对布局表的通性，以连通图的式分析输表结构以

持任意 FPGA 设计；

2) 加对卷积计算单元和矩阵计算单元的混合布局优化；

3) 研究细粒度布局，进步优化全局布局结果，使布线后的时钟频率更稳定。 47

基于进化算法的 FPGA 脉动阵列快速硬核布局法参考献

参考文献

[1] LUNDSTROM M. Moore’s law forever?[J]. Science, 2003, 299(5604) : 210 – 211.

[2] KIM N S, AUSTIN T, BAAUW D, et al. Leakage current: Moore’s law meets static power[J]. com- puter, 2003, 36(12) : 68 – 75.

[3] CHUNG E S, MILDER P A, HOE J C, et al. Single-chip heterogeneous computing: Does the future include custom logic, FPGAs, and GPGPUs?[C] // 2010 43rd annual IEEE/ACM international sympo- sium on microarchitecture. 2010 : 225 – 236.

[4] AMARA A, AMIEL F, EA T. FPGA vs. ASIC for low power applications[J]. Microelectronics journal, 2006, 37(8) : 669 – 677.

[5] SINGH G, CHELINI L, CORDA S, et al. Near-memory computing: Past, present, and future[J]. Mi- croprocessors and Microsystems, 2019, 71 : 102868.

[6] FAROOQ U, PARVEZ H, MEHREZ H, et al. Exploration of heterogeneous FPGA architectures[J].

International Journal of Reconfigurable Computing, 2011, 2011.

[7] SAMAJDAR A, GARG T, KRISHNA T, et al. Scaling the Cascades: Interconnect-aware FPGA im- plementation of Machine Learning problems[C] // 2019 29th International Conference on Field Pro- grammable Logic and Applications (FPL). 2019 : 1 – 8.

[8] LI W, DHAR S, PAN D Z. UTPlaceF: A routability-driven FPGA placer with physical and conges- tion aware packing[J/OL]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and

Systems, 2017, 37(4) : 869 – 882. http://wuxili.net/project/utplacef/.

[9] KIRKPATRICK S, GELATT C D, VECCHI M P. Optimization by simulated annealing[J]. science, 1983, 220(4598) : 671 – 680.

[10] GORT M, ANDERSON J H. Analytical placement for heterogeneous FPGAs[C] // 22nd international conference on field programmable logic and applications (FPL). 2012 : 143 – 150.

[11] MAIDEE P, ABABEI C, BAZARGAN K. Fast timing-driven partitioning-based placement for island style FPGAs[C] // Proceedings of the 40th annual design automation conference. 2003 : 598 – 603.

[12] LI K, ZHANG T, WANG R. Deep Reinforcement Learning for Multi-objective Optimization[J]. arXiv preprint arXiv:1906.02386, 2019. 49

基于进化算法的 FPGA 脉动阵列快速硬核布局法参考献

[13] LU Z, WHALEN I, BODDETI V, et al. Nsga-net: neural architecture search using multi-objective genetic algorithm[C] // Proceedings of the Genetic and Evolutionary Computation Conference. 2019 : 419 – 427.

[14] LOSHCHILOV I, HUTTER F. CMA-ES for hyperparameter optimization of deep neural networks[J]. arXiv preprint arXiv:1604.07269, 2016.

[15] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE transactions on evolutionary computation, 2002, 6(2) : 182 – 197.

[16] HANSEN N, OSTERMEIER A. Adapting arbitrary normal mutation distributions in evolution strate- gies: The covariance matrix adaptation[C] // Proceedings of IEEE international conference on evolu- tionary computation. 1996 : 312 – 317.

[17] SHEN Y, FERDMAN M, MILDER P. Maximizing CNN Accelerator Efficiency Through Resource Partitioning[C/OL] // ISCA '17 : Proceedings of the 44th Annual International Symposium on Com- puter Architecture. New York, NY, USA : ACM, 2017 : 535 – 547. http://doi.acm.org/10.1145/ 3079856.3080221.

[18] Shen Y, Ferdman M, Milder P. Overcoming resource underutilization in spatial CNN accelera- tors[C/OL] // 2016 26th International Conference on Field Programmable Logic and Applications (FPL). 2016 : 1 – 4. http://dx.doi.org/10.1109/FPL.2016.7577315.

[19] WU E, ZHANG X, BERMAN D, et al. Compute-Efficient Neural-Network Acceleration[C/OL] // Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Ar- rays, FPGA 2019, Seaside, CA, USA, February 24-26, 2019. 2019 : 191 – 200. https://doi.org/ 10.1145/3289602.3293925.

[20] Wu E, Zhang X, Berman D, et al. A high-throughput reconfigurable processing array for neural net- works[C/OL] // 2017 27th International Conference on Field Programmable Logic and Applications (FPL). 2017 : 1 – 4. http://dx.doi.org/10.23919/FPL.2017.8056794.

[21] VENKATRAMAN R, PATNAIK L M. An evolutionary approach to timing driven fpga placement[C] // Proceedings of the 10th Great Lakes symposium on VLSI. 2000 : 81 – 85.

[22] JAMIESON P. Revisiting Genetic Algorithms for the FPGA Placement Problem.[C] // GEM. 2010 : 16 – 22.

[23] JAMIESON P. Exploring inevitable convergence for a genetic algorithm persistent fpga placer[C] // Proceedings of the International Conference on Genetic and Evolutionary Methods (GEM). 2011 : 1. 50

基于进化算法的 FPGA 脉动阵列快速硬核布局法参考献

[24] JAMIESON P, GHARIBIAN F, SHANNON L. Supergenes in a genetic algorithm for heterogeneous FPGA placement[C] // 2013 IEEE Congress on Evolutionary Computation. 2013 : 253 – 260.

[25] PRAVEEN T, OTHERS. Multi-Objective Memetic Algorithm for FPGA Placement Using Parallel Genetic Annealing[J]. International Journal of Intelligent Systems and Applications, 2016, 8(4) : 60.

[26] SZEGEDY C, IOFFE S, VANHOUCKE V, et al. Inception-v4, inception-resnet and the impact of residual connections on learning[C] // Thirty-first AAAI conference on artificial intelligence. 2017.

[27] REDMON J, FARHADI A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

[28] BARKAU R L. UNET: One-dimensional unsteady flow through a full network of open channels. User's manual[R]. [S.l.] : Hydrologic Engineering Center Davis CA, 1996.

[29] QIU J, WANG J, YAO S, et al. Going deeper with embedded fpga platform for convolutional neural network[C] // Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. 2016 : 26 – 35.

[30] LAVIN C, KAVIANI A. Rapidwright: Enabling custom crafted implementations for fpgas[C] // 2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM). 2018 : 133 – 140.

[31] XILINX. Vivado Design Suite User Guide, Using Constraints (UG903)[K]. .

[32] PANCHAL G, PANCHAL D. Solving NP hard problems using genetic algorithm[J]. Transportation, 2015, 106 : 6 – 2.

[33] Eguro K, Hauck S. Simultaneous Retiming and Placement for Pipelined Netlists[C/OL] // 2008 16th International Symposium on Field-Programmable Custom Computing Machines. 2008 : 139 – 148. http://dx. doi.org/10.1109/FCCM.2008.21.

[34] YANG M, ALMAINI A, WANG L, et al. FPGA placement using genetic algorithm with simulated annealing[C] // 2005 6th International Conference on ASIC : Vol 2. 2005 : 806 – 810.

[35] WANG K, XU N. Ant colony optimization for symmetrical FPGA placement[C] // 2009 11th IEEE International Conference on Computer-Aided Design and Computer Graphics. 2009 : 561 – 563.

[36] MAIDEE P, ABABEI C, BAZARGAN K. Timing-driven partitioning-based placement for island style FPGAs[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2005, 24(3) : 395 – 406. 51

基于进化算法的 FPGA 脉动阵列快速硬核布局法参考献

[37] Gort M, Anderson J H. Analytical placement for heterogeneous FPGAs[C/OL] // 22nd International Conference on Field Programmable Logic and Applications (FPL). 2012 : 143 – 150. http://dx. doi.org/10.1109/FPL.2012.6339278.

[38] ABUOWAIMER Z, MAAROUF D, MARTIN T, et al. GPlace3. 0: Routability-driven analytic placer for UltraScale FPGA architectures[J]. ACM Transactions on Design Automation of Electronic Systems (TODAES), 2018, 23(5) : 1 – 33.

[39] SLOSS A N, GUSTAFSON S. 2019 Evolutionary Algorithms Review[J/OL]. CoRR, 2019, abs/1906.08870. http://arxiv.org/abs/1906.08870.

[40] LEE S-J, RAAHEMIFAR K. FPGA placement optimization methodology survey[C] // 2008 Canadian Conference on Electrical and Computer Engineering. 2008 : 001981 – 001986.

[41] FEIST T. Xilinx White Paper: Vivado Design Suite (WP416)[K]. .

[42] ANON. Opt4J - A Modular Framework for Meta-heuristic Optimization[C]. 2011 : 1723 – 1730.

[43] ROS R, HANSEN N. A simple modification in CMA-ES achieving linear time and space complexity[C] // International Conference on Parallel Problem Solving from Nature. 2008 : 296 – 305.

[44] LUU J, GOEDERS J, WAINBERG M, et al. VTR 7.0: Next Generation Architecture and CAD Sys- tem for FPGAs[J/OL]. ACM Trans. Reconfigurable Technol. Syst., 2014, 7(2) : 6:1 – 6:30. https://verilogtorouting.org/download/. 52

基于进化算法的 FPGA 脉动阵列快速硬核布局法相关的科研成果录

相关的科研成果目录

1. 发表论

[1] Niansong Zhang, Xiang Chen, Nachiket Kapre. RapidLayout: Fast Hard Block Placement of FPGA-optimized Systolic Arrays using Evolutionary Algorithms. Ac- cepted by 28th IEEE Annual International Symposium on Field-Programmable Cus- tom Computing Machines (FCCM). May 3 - May 6, 2020. (CCF-C)

2. 发明专利

(1) 张年崧，杨嵩毅，符顺，陈翔，基于计算机视觉成像的业型材何尺动检测法，No. 201811539019.8，2019 年 4 53

基于进化算法的 FPGA 脉动阵列快速硬核布局法致谢

致谢

时光飞逝，岁如梭，值此毕业之际我想曾经对指导我、帮助我、和励我的献上诚挚的谢意。

感谢我的母在学期间给与我的限持与帮助，你们是我坚强的后盾。

感谢陈翔师对我的培养，从我进您实验室的两年来直对我的悉指导和私帮助。回忆起在您指导下完成的第个 FPGA 卷积加速器项使我感慨良多。在两年的时间 ，您给了我许多学习和成长的机会， 励着我逐渐发现了的研究兴

趣，找到了未来的向。感谢您直以来未曾动摇地持和励我的追求，未来如果我取得学术上的任何成就，那是与您的付出密不可分。

感谢滑铁卢学的 Nachiket Kapre 教授对我细致微的指导和帮助，和对 Rapid-

Layout 项的巨付出。感谢 WatCAG 实验室的各位伙伴。

感谢清华学的汪师对我的持和帮助，感谢您在我访问学习的期间给了我很多励，也为我提供了实验的计算资源。感谢 NOVAUTO 公司的各位，在我忙于论的时候帮我分担作和安排时间。感谢 NICS 实验室的各位师兄师姐和同学，以及两位政师，给了我接纳关怀和学术作的指导。

感谢符顺师兄和陈志鸿师兄对我的指引，感谢 I3C 实验室的各位学长学姐给我的帮助。

感谢我的朋友们，感谢杨嵩毅同学和修圣杰同学在我成长学习过程中的帮助和陪伴。

最后我想感谢四年来学院对我的培养，感谢学院各位师对我学业上的指引与帮助，感谢路陪伴着我的同学们。

张年崧

2020 年 4 27 55

基于进化算法的 FPGA 脉动阵列快速硬核布局法附录 ANSGA-II 快速配排序附录 A NSGA-II 快速非支配排序算法 A.1: Fast-Non-Dominated-Sort(P)

1对于每个 p ∈ P 进行

2 Sp= n p= 0 对于每个 q ∈ P 进行

3如果 p q ; // If p dominates q

4则

5 Sp= S p∪ {q} ; // Add q to the set of solutions dominated by p

6否则如果 q p 则

7 np= n p+ 1 ; // Increase the domination counter of p

8如果 np=0 ; // p blones to the first front

9则

10 pr ank= 1 F

1= F

1∪ {p}

11 i = 1 ; // Initialize the front counter

12当 Fi = 进行

13 Q = ; // Used to store the members of the next front

14对于每个 p ∈ Fi进行

15对于每个 q ∈ Sp进行

16 nq= n q 1 如果 n q; // q belongs to the next front

17则

18 qr ank= i + 1 Q = Q ∪ {q}

19 i = i + 1 Fi= Q 57

基于进化算法的 FPGA 脉动阵列快速硬核布局法附录 BNSGA-II Crowd-Distance 排序附录 B NSGA-II Crowd-Distance 排序

先定义 Crowd-Comparison Operator:

Crowd-Comparison Operator (n) 在优化的多个进程中促进结果收敛到个均匀分

布的帕累托前端 (Pareto-Front)。假设种群中每个个体 i 有两个属性： nondomination rank (ir ank) crowding distance (id istance)

我们可以定义 Crowd-Comparion Operator n: i nj if (i rank< j rank) or ((ir ank= j rank) and (id istance> j distance))

也就是说，在两个 nondomination rank 不同的解中，我们更倾向于 rank 较低（较好）

的个。如果 rank 相同，则说明两个解在同个帕累托前端内，此时我们选择拥挤度

更低的个。

算法 B.1: Crowd-Distance-Assignment(I)

1 l = |I| ; // number of solutions in I

2对于每个 i 进行

3 set I[i]d istance= 0 ; // Initialize distance

4对于每个 objective m 进行

5 I = sort(I, m) ; // sort using each objective value

6 I[1]d istance= I[l] distance= inf ; // so that boundary points are always selected

7对于 i = 2 to (I 1) ; // for all other points

8进行

9 I[i]d istance= (I[i] distance+ I[i + 1].m I[i 1].m)/(fm ax m fm in m) 59

基于进化算法的 FPGA 脉动阵列快速硬核布局法附录 BNSGA-II Crowd-Distance 排序算法 B.2: Crowd-Dista nce-Sorting

1 Rt= P t∪ Q t; // combine parent and offsprint population

2 F = Fast-Non-Dominated-Sort(Rt) ; // F is all nondominated fronts of R t

3 Pt +1=

4 i = 1

5当 |Pt

+1| + |F i| <= N ; // until the parent population is filled

6进行

7 Crowding-Distance-Assignment(Fi)

8 Pt

+1= P t+1∪ F i; // include ith nondominated front in the parent pop

9 i = i + 1 ; // check the next front for inclusion

10 Sort(Fi, n) ; // sort in descending order using n

11 Pt

+1= P t+1∪ F i[1 : (N |P t+1|)] ; // choose the first (N |P t+1|) elements of Fi

12 Qt+ 1 = make-new-pop(P t+1) ; // use selection, crossover, and mutation to create a new poputlation

13 t = t + 1 ; // increment the generation counter 60

毕业论文 (设计) 成绩评定记录指导教师评语：

论写作规范，引献充分，符合中学本科论的规范，

是篇优秀/良好/中等/合格的论 。

成绩评定：

指导教师签名: 年答辩组或专业负责意见：

成绩评定：

签名: 年院系负责意见：

成绩评定：

签名（章）： 年