



RAPID LAYOUT:

基于进化算法的脉动阵列快速硬核布局方法

指导老师：陈翔 副教授 Nachiket G. Kapre 副教授

答辩： 张年崧

2020 年 5 月 10 日

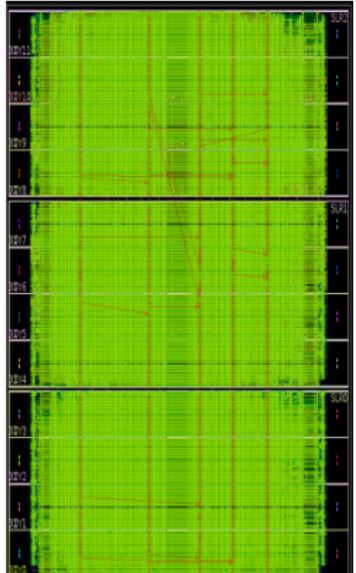
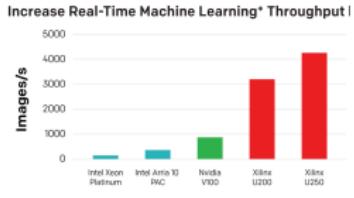
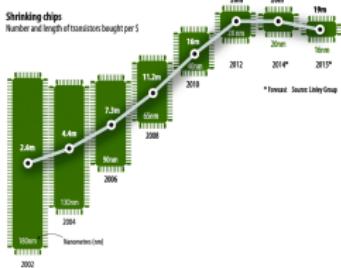
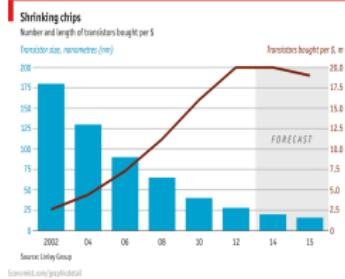
中山大学电子与信息工程学院通信工程专业

Table of Contents

1. 背景介绍
2. 基于进化算法的脉动阵列布局
3. 实验结果与分析
4. 总结与展望
5. 附录

背景介绍

选题背景



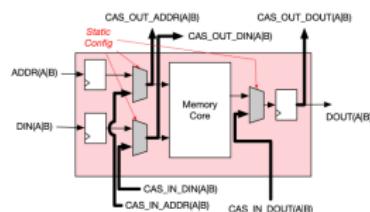
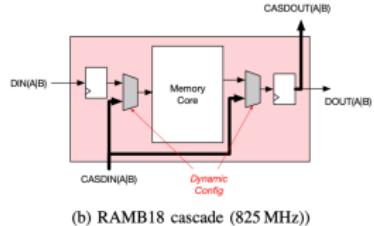
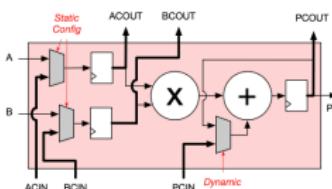
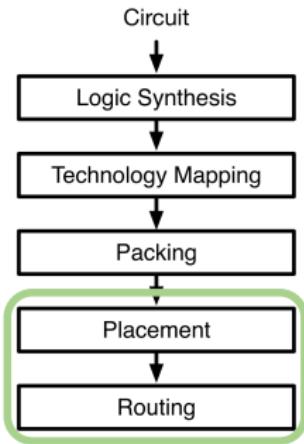
(a) 摩尔定律逐渐失效

(b) ALVEO 数据中心加速卡

(c) EDA 工具无法满足需求

Figure 1: “后摩尔时代”，FPGA 应用广泛，但 EDA 工具无法适应大规模设计

FPGA 实现过程与硬核 IP 介绍



现有布局算法：

1. 模拟退火：性能好但时间长，伸缩性差
2. Min-Cut 法：速度快但易收敛至局部最优
3. 解析布局算法：SOTA 但合法化复杂
4. 进化算法：性能次于退火，适用并行化

脉动阵列与进化算法

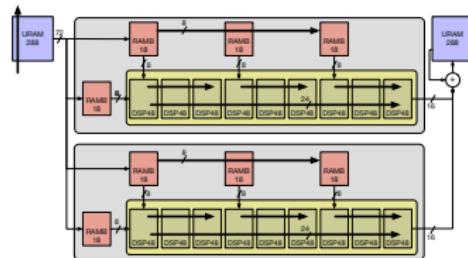
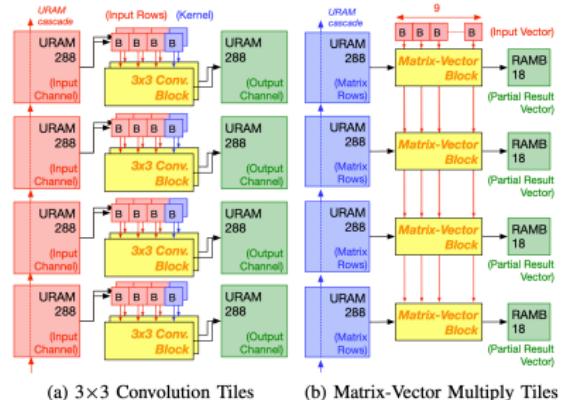


Figure 2: 脉动阵列设计

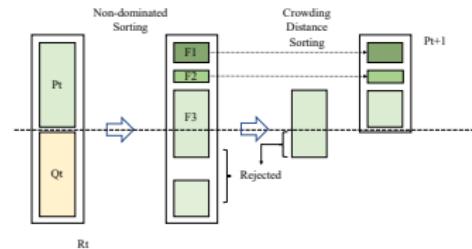
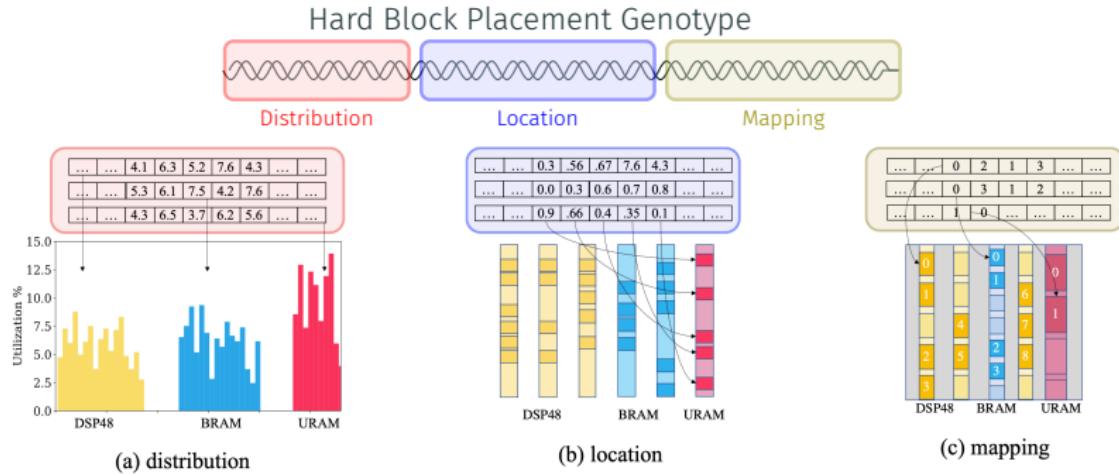


Figure 3: CMA-ES 与 NSGA-II

基于进化算法的脉动阵列布局

问题建模与基因型设计



Minimize:

$$\sum_{i,j} ((\Delta x_{i,j} + \Delta y_{i,j}) \cdot w_{i,j})^2 \quad (1)$$

$$\max_k (BBoxSize(C_k)) \quad (2)$$

subject to:

$$0 \leq x_i < XMAX, 0 \leq y_i < YMAX \quad (3)$$

$$x_i, y_i \neq x_j, y_j \quad (4)$$

如果 i 级联在 j 之后: $x_i = x_j$

$$y_i = \begin{cases} y_j + 1 & i, j \in \{DSP, URAM\} \\ y_j + 2 & i, j \in \{RAMB\} \end{cases} \quad (5)$$

RapidLayout 工作流

A 逻辑网表复制：

输入单个卷积单元逻辑网表，自动确定布局的最小可复制矩形区域和最大容纳单元数，复制逻辑网表。

B 基于进化算法硬核布局：

使用 NSGA-II 或 CMA-ES 求解。

C 物理网表布局和 Site 布线：

更新物理网表，连接 Site 内信号。

D 流水线：

自动确定流水线级数，优化时钟。

E 超逻辑域布局与布线：

在 SLR 内完成细节布局和布线。

F 超逻辑域复制：

重用布线结果，将实现后的设计复制到整个 FPGA。

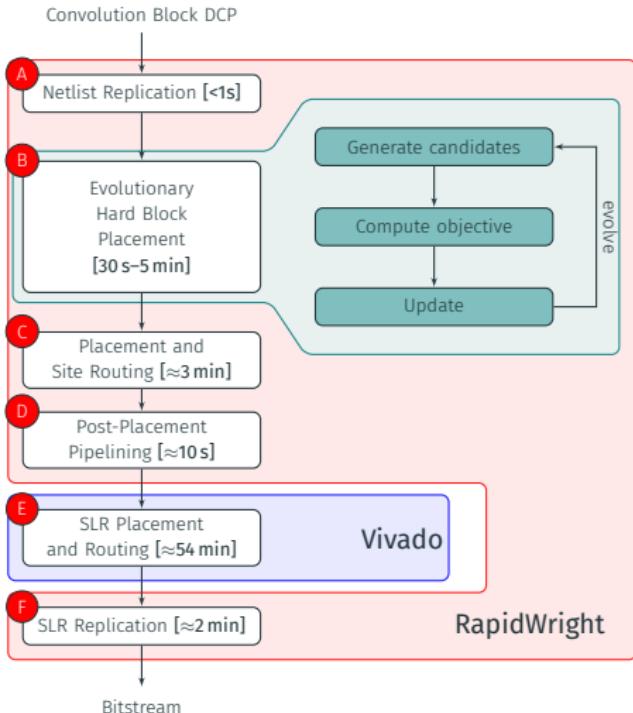


Figure 4: RapidLayout 工作流

以 Xilinx UltraScale+ VU11P 为例的实现流程

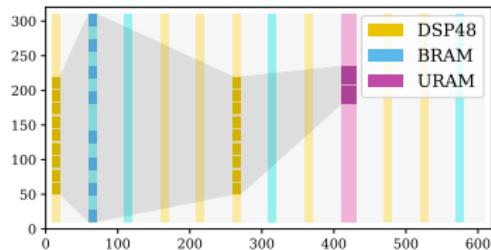


Figure 5: 单卷积单元布局结果

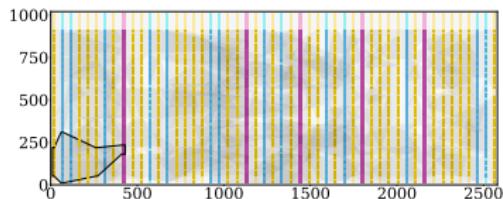


Figure 6: Repeat Rect. 布局结果-80 核

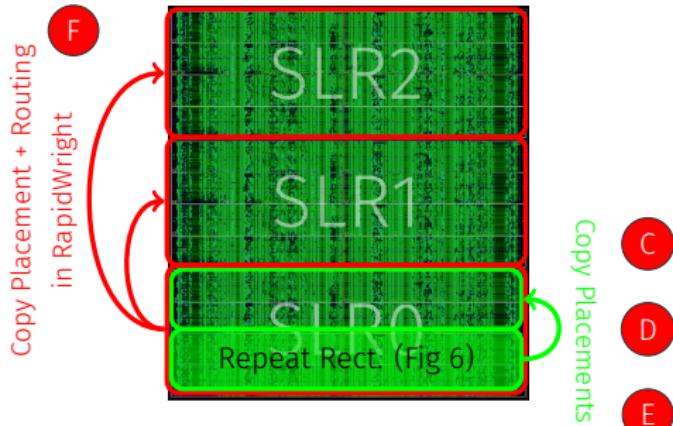


Figure 7: 布局重用与 SLR 布线重用

- 布局和布线结果重用，RapidLayout 端到端实现仅需 1 小时，比 Vivado 快 5-6 倍；
- 自动化方法免去了手工布局和试错；
- 优质布局结果与自动流水线确保时钟频率达到 URAM 上限 650 MHz。

实验结果与分析

实验设计与算法性能对比

- 参数优化的模拟退火；
- 标准布局工具 VPR；
- 最先进解析布局 UTPlaceF；
- 单目标遗传算法（仅优化边界框）；
- 手工布局设计。

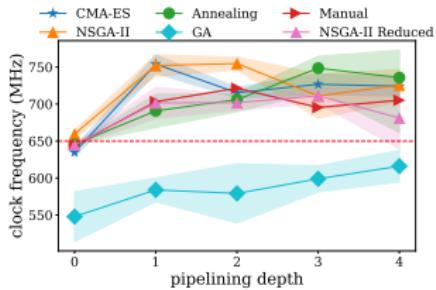


Figure 8: 时钟频率 vs 流水级数

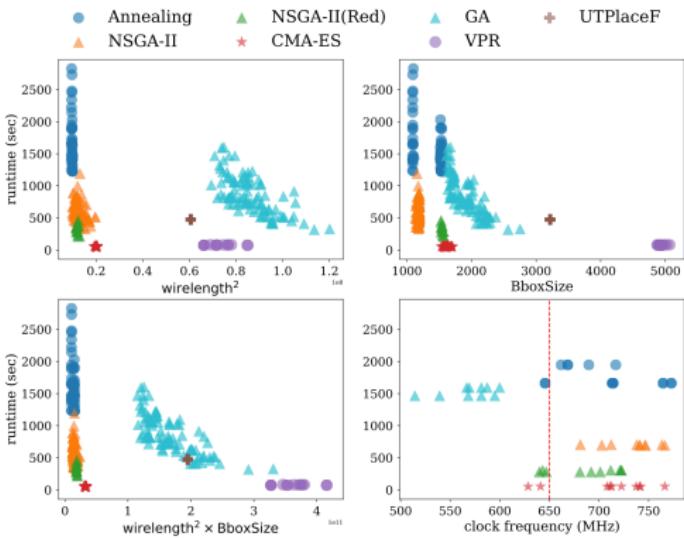


Figure 9: 线长、边界框尺寸、时钟频率对比

实验结论:

1. 进化算法相对 SA 快 2.7–30.8 倍，线长相对 UTPlaceF, VPR 优化 1.8–2.4 倍；
2. 边界框尺寸相对二者优化 2.0–4.1 倍；
3. 进化算法 1–2 级流水可达 750MHz 超高频率，NSGA-II 节省约 17K(6%) 流水寄存器。

迁移学习性能

Table 1: 迁移学习性能: VU3P, VU11P 作为初始设备

Device	Design Size (conv units)	Impl.Runtime (mins.)	Frequency (MHz)	Placement Runtime	
				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×)

- 同设计在不同 FPGA 的硬核布局可进行迁移学习，仅需满足硬核列数相同；
- 迁移学习可加快优化进程7–12倍。
- 迁移学习获得结果的时钟频率相对无迁移学习变化 -2% – +7%

总结与展望

总结与展望

本毕业设计的主要贡献为：

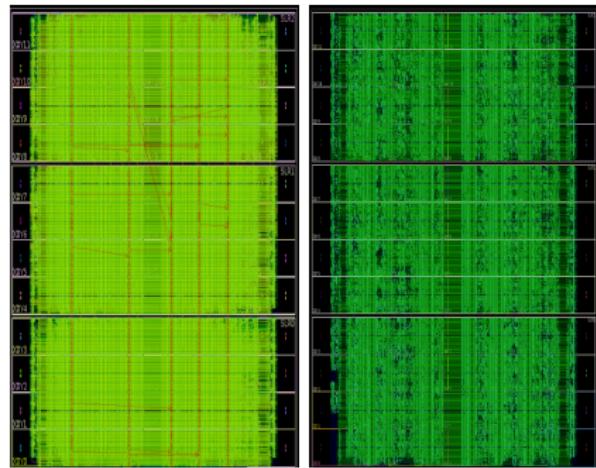
1. 提出时间更短、性能更优的进化布局算法，解决大规模加速器设计布局难度大、时间成本高的问题；
2. 提出布局结果的迁移学习方法，构建 RapidLayout 实现框架。

研究展望：

1. 通用性：支持任意硬核设计布局；
2. GPU 加速：更大规模的并行加速。

相关科研成果：

1. N. Zhang, X. Chen, N. Kapre. *RapidLayout: Fast Hard Block Placement of FPGA-optimized Systolic Arrays using Evolutionary Algorithms*, International Conference on Field-Programmable and Application (FPL 2020, CCF-C, 15.2% Acceptance Rate)
2. 张年崧, 杨嵩毅, 符顺, 陈翔, 基于计算机视觉成像的工业型材几何尺寸自动检测方法, No. 201811539019.8, 2019 年 4 月



(a) Vivado

(b) RapidLayout

Figure 10: 480 核 CNN 加速器布局布线

附录

异构 FPGA 硬核列分布资源

