



上海科技大学
ShanghaiTech University

硕士学位论文

基于阻变存储器的神经网络加速器的供电可靠性以及热敏可靠性研究

作者姓名: 张呈瑞

指导教师: 周平强 副教授

上海科技大学信息科学与技术学院

学位类别: 工学硕士

一级学科: 计算机科学与技术

学校/学院名称: 上海科技大学信息科学与技术学院

2022 年 06 月

Research on The Power Supply and Thermal Reliability
of ReRAM Based Neural Network Accelerator

A thesis submitted to
ShanghaiTech University
in partial fulfillment of the requirement
for the degree of
Master of Science in Engineering
in Computer Science and Technology
By
Zhang Chengrui
Supervisor: Professor Zhou Pingqiang

School of Information Science and Technology
ShanghaiTech University

06 / 2022

上海科技大学
研究生学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师的指导下独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明或致谢。

作者签名：

日 期：

上海科技大学
学位论文授权使用声明

本人完全了解并同意遵守上海科技大学有关保存和使用学位论文的规定，即上海科技大学有权保留送交学位论文的副本，允许该论文被查阅，可以按照学术研究公开原则和保护知识产权的原则公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名：

导师签名：

日 期：

日 期：

摘要

阻变存储器可以提高芯片集成度，以及改进芯片计算架构。同时阻变存储器独特的结构也使其可以通过存内计算的方式加速神经网络中的向量-矩阵运算。因此基于阻变存储器的神经网络加速器（下称加速器）成为当前的研究热点。

由于该加速器仍存在可靠性问题，且随着集成度的上升，对这些问题进行分析和改进的难度也在不断加大。因此可靠性问题成为该加速器的相关研究中最重要的问题之一。本文主要对该加速器芯片的供电网络可靠性以及阻变存储器的热敏可靠性问题进行研究。

针对加速器芯片中的供电网络可靠性问题，当前的难点主要在于如何加速供电网络的分析。本文针对该芯片上的静态供电网络分析方法进行了优化和加速。本文首先扩展了传统的层级结构式供电网络分析方法，使其可以在有电压源的局部电路上进行分析。其次，针对当前加速器芯片中多核同构的特性，本文提出针对多核同构芯片的供电网络分析方法。实验结果表明，本文提出的方法可以在 8 核同构的加速器芯片上达到 7.17 倍的分析加速以及 99.9% 的精度，且该加速倍数与同构核数成正比关系。

针对加速器芯片中由于高功耗密度导致的阻变存储器热敏可靠性问题，当前的难点主要在于如何降低阻变存储器交叉阵列（下称交叉阵列）上的峰值功耗密度。为达到这个目标，本文针对性地提出了结构调整方法和改进的交叉阵列权重排布方法。这两种方法分别通过优化加速器中高功耗元件的布局，以及改进神经网络权重到交叉阵列的映射方法来降低芯片峰值功耗密度。实验结果表明，本文所提出的方法可在 VGG11 和 VGG9 网络上分别降低 5.0K 和 10.4K 的加速器芯片峰值温度，以及分别延长 1.3 倍和 1.72 倍的加速器使用寿命。

关键词： 阻变存储器，神经网络加速器，可靠性，供电网络分析

Abstract

Resistive memory (ReRAM) can improve the integration level of chips as well as the computing architecture. Besides, the unique structure of ReRAM also enables it to accelerate vector-matrix operations in the neural network through In-memory Computing. ReRAM-based neural network accelerators (hereinafter referred to as accelerators) have therefore become a current research hotspot.

Since the accelerator still has reliability problems, and as the level of integration increases, it becomes increasingly difficult to analyze and improve these problems. Therefore, the reliability problem has become one of the most important issues in the related research of this accelerator. This paper mainly studies the reliability of the power distribution network (PDN) of the accelerator chip and the thermal reliability of the ReRAM.

For the reliability of the PDN in the accelerator chip, the current difficulty lies in how to speed up the analysis of the power supply network. This thesis optimizes and accelerates the static PDN analysis method on this chip. In this thesis, the traditional hierarchical PDN analysis method is firstly extended so that it can be analyzed on partial circuits with voltage sources. Secondly, according to the characteristics of multi-homogeneous cores in current accelerator chips, this thesis proposes a PDN analysis method for them. The experimental results show that the method proposed in this thesis can achieve 7.17 times the analysis speedup and 99.9% accuracy on the 8-core homogeneous accelerator chip, and the speedup factor is proportional to the number of homogeneous cores.

Aiming at the thermal reliability problem of ReRAM in accelerator chips which is caused by high power consumption density, the current difficulty mainly lies in how to reduce the peak power consumption density on ReRAM crossbar arrays (hereinafter referred to as crossbar arrays). In order to achieve this goal, this thesis proposes a structure adjustment method and an improved cross-array weight arrangement method. These two approaches reduce chip peak power density by optimizing the placement of high-power components in the accelerator and improving the mapping of neural network weights

to crossbar arrays, respectively. The experimental results show that the method proposed in this thesis can reduce the peak temperature of accelerator chips by 5.0K and 10.4K on the VGG11 and VGG9 networks, respectively, and prolong the endurance of the accelerator by 1.3 times and 1.72 times, respectively.

Key Words: ReRAM, neural network accelerator, reliability, power grid analysis

目 录

第1章 绪论	1
1.1 失效的登纳德缩放定律和摩尔定律	1
1.2 芯片的发展趋势	3
1.2.1 发展趋势概述.....	3
1.2.2 超越摩尔	4
1.2.3 阻变存储器应用概述	4
1.3 基于阻变存储器的加速器芯片的可靠性问题	6
1.3.1 电路可靠性问题	7
1.3.2 阻变存储器的可靠性问题.....	8
1.4 本文关注的问题	8
1.5 本文的组织结构	10
第2章 基于阻变存储器的神经网络加速器芯片概述	11
2.1 神经网络	11
2.1.1 人工神经网络.....	12
2.1.2 脉冲神经网络.....	13
2.2 基于阻变存储器的神经网络加速器芯片	15
2.2.1 阻变存储器	15
2.2.2 神经网络加速器的芯片架构	16
2.3 加速器芯片中阻变存储器的可靠性问题	19
2.3.1 可靠性问题概述	19
2.3.2 阻变存储器的热敏问题	20
2.4 加速器芯片中供电网络可靠性问题	21
2.4.1 供电网络概述	21
2.4.2 层级式供电网络分析方法.....	21
2.5 本章小结	23
第3章 针对加速器芯片电路的供电网络分析加速算法	25
3.1 观察与动机	26
3.2 对层级式供电网络分析方法的改进	27
3.3 对同构芯片的加速分析方法	28
3.4 实验结果	30
3.4.1 实验设置	30
3.4.2 实验结果分析.....	30
3.5 本章小结	33

第 4 章 针对加速器芯片中交叉阵列的热敏可靠性研究 ······	35
4.1 动机与解决思路 ······	36
4.1.1 加速器芯片的布局 ······	36
4.1.2 输入分布对功耗的影响 ······	37
4.1.3 针对权重排布算法的优化 ······	39
4.1.4 方法流程 ······	40
4.2 结构调整 ······	41
4.3 功耗分布 ······	42
4.3.1 初始热图构建 ······	43
4.3.2 对输入的分析 ······	43
4.3.3 基于输入的权重排布方法 ······	44
4.4 实验结果 ······	48
4.4.1 实验设置 ······	48
4.4.2 实验结果及分析 ······	49
4.5 本章小结 ······	52
第 5 章 总结与展望 ······	54
5.1 总结 ······	54
5.2 展望 ······	55
参考文献 ······	57
作者简历及攻读学位期间发表的学术论文与研究成果 ······	63

图形列表

1.1 摩尔定律的趋势与实际晶体管密度的区别。	2
1.2 单位面积芯片的功耗与晶体管技术节点的关系 (Hennessy 等, 2019)。	3
1.3 ITRS 对新型存储的评估 (Chen, 2016)。	5
1.4 基于阻变存储器的神经网络加速器示例 (Joardar 等, 2019)。	6
1.5 暗硅现象示意图 (Goulding 等, 2010)。	7
2.1 神经元结构 (Wikipedia, n.d.)。	11
2.2 人工神经网络结构 (Abiodun 等, 2018)。	12
2.3 常见激活函数示例。	13
2.4 (a) 低阻态 (LRS) 下的阻变存储器。(b) 高阻态 (HRS) 下的阻变存储器。 (Ambrosi 等, 2019).....	15
2.5 (a) 基于 HfO_2 器件的阻变存储器 I-V 曲线。(b) 基于 SiO_x 器件的阻变存储器 I-V 曲线。 (Ambrosi 等, 2019)	16
2.6 用于神经网络加速的 1T1R 交叉阵列 (Yao 等, 2017)。	17
2.7 ISAAC 架构 (Shafiee 等, 2016)。	18
2.8 ISAAC 的工作流程图 (Shafiee 等, 2016)。	19
2.9 热效应作用下阻变存储器的电流-电压曲线 (C. Walczyk 等, 2011)。 ..	20
2.10 热效应作用下阻变存储器电导值的变化曲线 (Beigi 等, 2018a)。	21
2.11 层级式供电网络分析流程 (Zhao 等, 2002)。	23
3.1 GPU 的计算架构 (Nickolls 等, 2010)。	26
3.2 电路实例。节点 1 连接电压源, 节点 4 连接电流源, I 表示外部电流。	27
3.3 GMRES 算法流程图 (X. Ma, 2013)。	29
3.4 加速结果展示图。	33
4.1 脉冲神经网络从第 i 层到第 $i + 1$ 层的推断过程。	37
4.2 VGG11 网络的第二层全连接层在训练集和测试集上归一化后的输入分布。	39
4.3 对权重矩阵进行不同的列排布方法的差异。	40
4.4 对交叉阵列功耗 (热) 问题的优化框架。	41
4.5 (a)ISAAC 结构。(b) 改进后的 IMA 结构。	42
4.6 交叉阵列的初始热图。	43
4.7 改进的权重排布算法。	47

4.8 BLDM 算法示例.....	48
4.9 (a) 标准化的功率范围, (b) 卷积和全连接层的功率范围减少的幅度。.....	50
4.10 映射到加速器芯片的 <i>VGG9</i> 网络中最热的 TILE 的温度分布	51
4.11 VGG11 和 VGG9 在不同方法下的 (a) 最高温度比对图。 (b) 标准化后的 使用寿命比对图。	51
4.12 阻变存储器的使用寿命随温度的变化曲线 (Beigi 等, 2018b)。	52

表格列表

3.1 IHA 的分析在 I 型同构芯片上的分析结果。	31
3.2 IHA 的分析在 II 型同构芯片上的分析结果。	32
3.3 IHA 的分析在 III 型同构芯片上的分析结果。	32
4.1 ISAAC 中部分参数。	49

符号列表

符号	说明	单位
R	the resistance	Ω
G	the conductance	S

缩写

缩写	全称
CPU	Central Processing Unit
GPU	General Purpose Graphics Processing Units
MAC	Multiply accumulate
RRAM	Resistive Random Access Memory
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
SNN	Spiking Neural Network
MCA	Memristor Crossbar Array
IF	Integrate-and-fire
MP	Membrane potential
ADC	Analog to digital converter
DAC	Digital to analog converter

算子 & 说明

Δ	difference
∂	partial difference

第1章 绪论

随着芯片制造工艺的不断进步，单位面积芯片上可集成的晶体管数量不断上升。然而，由于晶体管尺寸不断逼近物理极限，制造工艺的进步反而给芯片的应用带来一些问题，如暗硅现象等。类如阻变存储器等新兴器件的诞生为解决这些问题带来希望。因此，如何优化并使用这些新兴器件成为当前学术界的研究热点。

本章首先对当前芯片工艺的技术瓶颈进行简单介绍，然后阐述当前芯片发展的三个主要趋势，由此引入基于阻变存储器的神经网络加速器芯片（下称加速器）。此后，介绍目前在该加速器的设计及使用过程中存在的可靠性问题。最后将介绍本文的主要工作和文章的组织框架。

1.1 失效的登纳德缩放定律和摩尔定律

晶体管的诞生至今已有七十多年，自 1947 年，美国贝尔实验室（Alcatel-Lucent Bell Labs）的肖克利、巴丁和布拉顿组成的研究小组，研制出一种点接触型的锗晶体管后，半导体行业在无数研究者的努力下蓬勃发展。1958 年，美国德州仪器公司（Texas Instruments）生产出世界上第一块集成电路 (Ayers, 2018)，自此，集成电路行业开始了飞速发展。此外，在上世纪六十年代和七十年代中，戈登摩尔先生和罗伯特登纳德先生分别对半导体工业的发展进程做出了预言，而这两条预言也在接下来的近 50 年里成为了半导体工业发展的准则：

- 摩尔定律（Moore's Law）(Moore, 1998)：在同样面积的芯片上所能集成的晶体管的数目，每 18-24 个月翻一倍，同时，芯片的性能提高一倍，而价格下降一半。
- 登纳德缩放定律（Dennard Scaling）(Dennard 等, 1974)：晶体管的尺寸在每一代技术中都将缩小 30% (0.7 倍)，因此它们的面积会减少 50%。这意味着电路的延迟减少 30% (0.7 倍)，因此增加了约 40% (1.4 倍) 的工作频率。最后，为保持电场恒定，电压降低了 30%，能量降低了 65%，功率降低了 50%。因此，在每一代技术中，晶体管密度增加一倍，电路速度提高 40%，但相同面积的芯片上

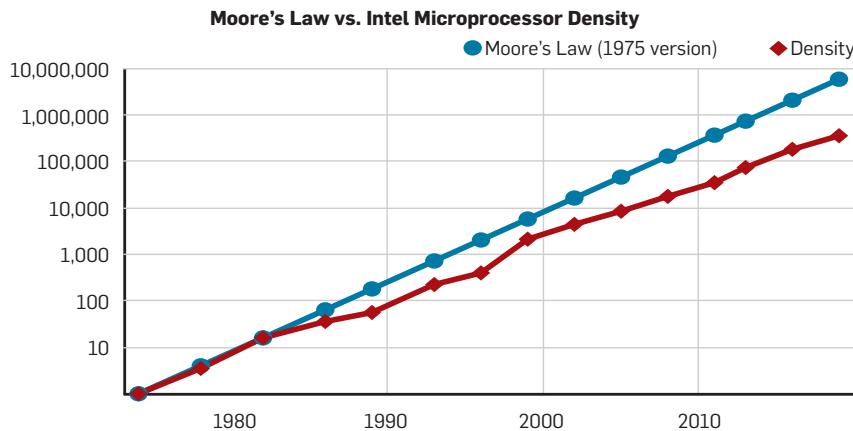


图 1.1 摩尔定律的趋势与实际晶体管密度的区别。

Figure 1.1 Trends of transistor density of Moore' s law and reality.

的功耗(晶体管数量增加一倍)保持不变。

这两条定律在其提出后的数十年内,指导了半导体行业和芯片行业中在晶体管的密度,芯片的性能和功耗方面的进步。图 1.1 描述了自上世纪七十年代起,摩尔定律和同时期英特尔公司生产的微处理器的片上晶体管数目的对比。从图中可以看出,在摩尔定律刚提出后的近十年内,实际芯片的发展趋势和预测基本一致,而随着晶体管集成度的不断上升,技术的演进逐渐无法跟随摩尔定律。在 21 世纪里,由于晶体管制造工艺不断逼近物理极限($7\text{nm} \rightarrow 5\text{nm} \rightarrow 3\text{nm}$),漏电流,量子效应等负面作用(Kim, 2010)使得晶体管的尺寸难以进一步减少。因此,摩尔定律在 21 世纪已逐渐失效(QUARTERLY, 2016),芯片的集成度进步很难再吻合摩尔定律的预测。

除摩尔定律的失效以外,登纳德缩放定律在 21 世纪初也被宣布失效(Esmailzadeh 等, 2011)。图 1.2 描述了 21 世纪以来晶体管的工艺制程和单位面积芯片的功耗的关系,从图中可以看出,在工艺制程在 2006 年达到 65nm 后(HUAWEI, 2020),芯片上的单位功耗便无法如登纳德缩放定律描述的一样随工艺的上升而保持不变,而随着工艺制程尺寸的不断减少,芯片的单位面积功耗相对 2006 年已增加了约 8 倍。

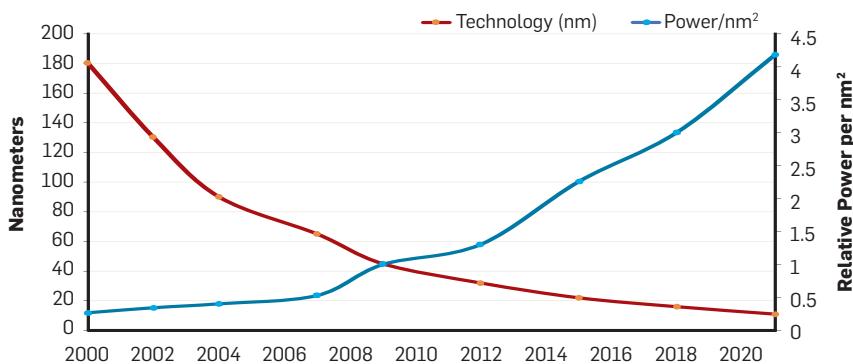


图 1.2 单位面积芯片的功耗与晶体管技术节点的关系 (Hennessy 等, 2019)。

Figure 1.2 Power per nm^2 vs technology node (Hennessy 等, 2019).

1.2 芯片的发展趋势

1.2.1 发展趋势概述

摩尔定律和登纳德缩放定律的失效意味着现有的芯片技术难以如上世纪一样快速发展，因此，在 21 世纪初期，国际半导体技术发展路线图（International Technology Roadmap for Semiconductors）提出了以下三个可能的发展方向 (Khan 等, 2018)：

- 延续摩尔 (More Moore): 延续摩尔基本思路是从传统互补金属氧化物半导体 (Complementary Metal Oxide Semiconductor, 简称 CMOS) 转向非传统 CMOS，包括半节距按比例减小，采用非经典器件结构等，从结构的设计及布局创新来实现芯片的微缩，其本质是通过采用新的器件结构和布局来实现芯片的设计和加工。
- 扩展摩尔 (More than Moore): 与延续摩尔所采用的方式不同，扩展摩尔的本质是将不同功能的芯片和元件组装拼接在一起封装。其创新点在于异构的封装技术，在满足需求的情况下，可快速和有效的实现芯片功能，具有设计难度低、制造便捷和成本低等优势。
- 超越摩尔 (Beyond Moore): 新兴器件是超越摩尔领域取得突破的关键。超越摩尔的本质是使用新兴器件，使其减少信号在电路中传递所消耗的能量并提升电路性能。其动机是，在集成电路目前的架构中，信息的传递和处理都是以电子作为基本单元。而从信息传递的角度来看，单个电子是不能传递信息的，多电子组合才能携带信息。与此同时，信号在传递过程中还会存在能量消耗并

产生热量。若寻找到其他基本单元自身可以携带信息或者信息传递过程中不会消耗能量，将会降低功耗并提升性能，打破现在所面临的发展瓶颈问题 (Zhirnov 等, 2010)。目前超越摩尔方向主要处在研究阶段，阻变存储器、铁电存储器、相变存储器等能够实现自组装的器件是超越摩尔方向研究的热点。

延续摩尔和扩展摩尔两种方向依然在 CMOS 工艺下进行研究探索，如 21 世纪初以来，处理器设计者对多核芯片 (Esmaeilzadeh 等, 2011)，以及 3D 封装工艺 (Karnezos, 2004) 的倾向，然而，由于它们仍基于 CMOS 工艺，实践证明，它们的性能均受到了摩尔定律的制约 (Sutter 等, 2005)。而在超越摩尔中，新兴的器件在提高芯片集成度，降低芯片功耗，以及提升芯片架构方面均具有优势 (X. Liu 等, 2016; Zhirnov 等, 2010)，因此，本文主要对超越摩尔的芯片的分析研究。

1.2.2 超越摩尔

新型器件可以提高芯片集成度，降低芯片功耗，以及改进传统的芯片架构 (X. Liu 等, 2016; Zhirnov 等, 2010)。因此，近年来新兴器件得到了广泛的研究，目前比较主流的几种器件是：阻变随机存储器 (Resistive RAM，简称 memristor、ReRAM 或 RRAM，下称阻变存储器)、铁电随机存储器 (Ferroelectric RAM，简称 FeRAM)、相变存储器 (Phase change memory，简称 PCM)、以及自旋转移转矩磁性随机存储器 (Spin-transfer-torque magnetic RAM，简称 STTMRAM) 等。这些新型器件相比于传统的 CMOS 器件来说在耐久度，带宽，面积以及写入速度等方面各具优势 (Amirsoleimani 等, 2020; Chen, 2016; Xue 等, 2011)。

在这些器件中，阻变存储器由于其结构简单和成本低的优势，近年来受到学术界的青睐。此外，如图 1.3 所示，在 ITRS 的一份调研里 (Chen, 2016)，阻变存储器由于其综合指标的均衡性，在图示的六种新型器件 (FeFET-铁电场效应管、RRAM-阻变存储器、Mott-莫特存储器、Macromolecular-高分子存储器、Molecular-分子存储器以及 Carbon-碳基存储器) 中脱颖而出。因此，本文主要针对基于阻变存储器的芯片进行研究。

1.2.3 阻变存储器应用概述

阻变存储器作为一种新型的非易失(Non-volatile)存储器，具有并行高速、低功耗以及大容量等特点 (Wong 等, 2012)。其不仅可以实现对数据的高密度存储

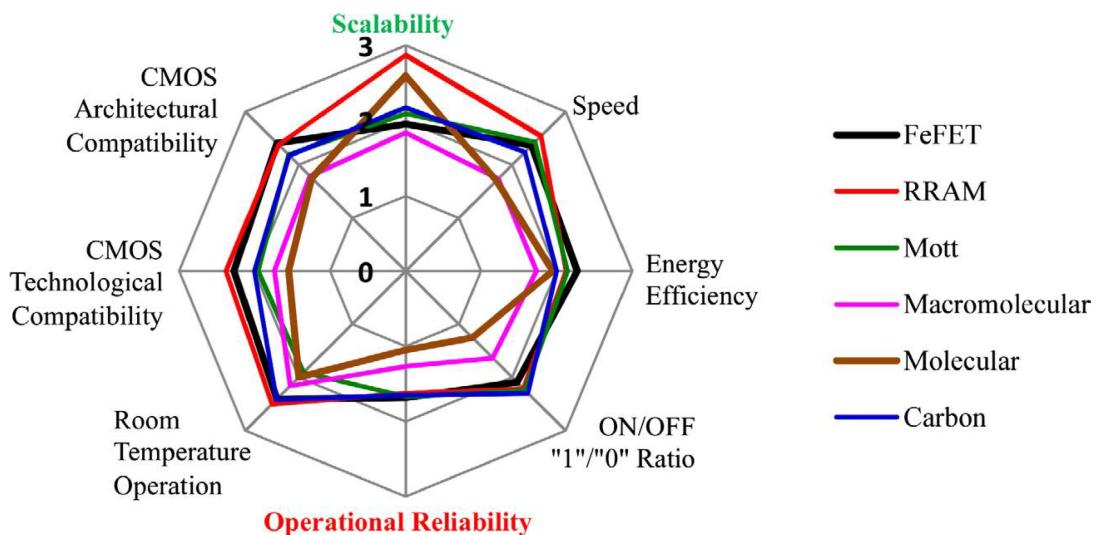


图 1.3 ITRS 对新型存储的评估 (Chen, 2016)。

Figure 1.3 ITRS critical review of emerging memories (Chen, 2016).

(每个单元 $4\text{-}12F^2$ (Yu, 2018)), 还可以打破传统冯诺依曼架构的内存墙 (Memory wall (Wulf 等, 1995)) 的限制, 实现存内计算 (Processing in memory, 简称 PIM)。存内计算是一种兼顾存储与计算的新型计算架构, 其计算的过程与冯诺依曼结构不同, 不需要将数据在存储单元和计算单元之间移动 (访存, 执行, 写回), 而可以利用其交叉阵列 (Crossbar, 简称 Xbar 或 XB) 的存储结构, 将部分数据直接在内存中进行计算。这种方式不仅可以减少数据的传输时间, 还可以利用基尔霍夫定律在内存中快速完成向量与矩阵的乘法操作 (Vector-matrix multiplication), 通过这种方式, 向量矩阵乘法的复杂度将从 $O(n^2)$ 降低至 $O(1)$ 。因此, 相对于冯诺依曼结构中的向量矩阵乘法操作, 该结构的计算效率更高。

此外, 由于在 21 世纪中, 人工智能, 特别是神经网络得到了大幅的发展的广泛的应用, 而在神经网络的计算过程中, 向量矩阵乘法又占据主要部分 (陈煌等, 2018)。因此, 近年来基于阻变式存储器的神经网络加速器 (下称加速器) 得到了学术界和工业界的深入研究。本文也专注于对该加速器的研究。

比较知名的加速器的架构设计有两种, 一种是 ISAAC 结构 (Shafiee 等, 2016), 其将阻变存储器作为计算单元, 通过设计出适配其工作状态的新芯片架构来更好的应用其计算功能; 另一种则是 PRIME 结构 (Chi 等, 2016), 其结构更为通用, 既保留了阻变存储器的计算功能, 也保留了其作为普通存储器件的存储功能。

本文专注于 ISAAC 结构, 在该结构中, 其使用基于阻变存储器的交叉阵列

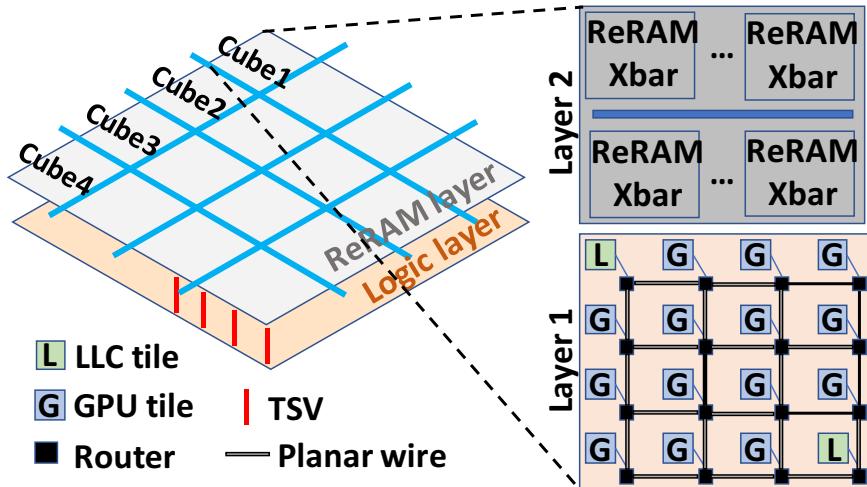


图 1.4 基于阻变存储器的神经网络加速器示例 (Joardar 等, 2019)。

Figure 1.4 An example of ReRAM based neural network accelerator (Joardar 等, 2019).

作为计算单元，来实现对权重的存储和对向量矩阵乘法的计算。在 ISAAC 结构中，其假定神经网络每一层的输入和每一个权重矩阵的分辨率均是 16 比特，以及每一个阻变存储器的分辨率为 2 比特。每一次使用 ISAAC 做乘加运算时，其将输入向量分为 16 个时钟周期，依次送入至交叉阵列中，将每个周期得到的结果存放至输出寄存器，并通过不断移位累加获得最终的计算结果。图 1.4 展示了目前学术界对于 ISAAC 结构的一个理想设计图 (Joardar 等, 2019)，其将芯片分为两个部分，第一个部分是传统的 GPU 核 (Logic layer)，第二个基于阻变存储器的交叉阵列群 (ReRAM layer)，以此来实现传统 CMOS 工艺和新型器件的结合，其 GPU 核多用于传统的逻辑推断，而交叉阵列群则用于加速神经网络的计算过程。

1.3 基于阻变存储器的加速器芯片的可靠性问题

基于阻变存储器的芯片是当前集成电路行业发展的主流方向之一，且其神经网络加速器的设计应用也与当前人工智能的时代潮流对应。然而，由于其本身的可靠性并不如传统 CMOS 工艺下的器件，因此，目前学术界仍致力于提高阻变存储器的可靠性或对因可靠性造成的问题进行在线或者离线补偿 (Shin 等, 2020)。除此之外，随着芯片集成度的不断提升和登纳德缩放定律的失效，该芯片在 CMOS 工艺下的电路器件也逐渐出现可靠性问题。

1.3.1 电路可靠性问题

在加速器芯片中，除含阻变存储器的交叉阵列部分，其余部分（辅助交叉阵列计算的电路，ReRAM 层外的逻辑层等）仍基于传统的 CMOS 工艺。因此，其性能也受到 CMOS 工艺的制约。而在 CMOS 工艺下，芯片集成度的上升带来的分析以及可靠性问题主要分为两个方面：

- 集成度上升带来的芯片分析难度上升问题：芯片的生产过程主要分为设计-验证-制造-封装-交付这五个步骤 (Bushnell 等, 2004)，其中芯片设计者主要参与的是芯片的设计与验证部分。芯片在设计之后会经历一系列的测试，去测试其性能、可靠性以及寿命等指标，常见的测试方法包括参数型测试（如延迟测试，开路短路测试等）以及功能性测试（通过输入参数判断芯片是否满足特定指标，如供电网络分析（Power grid analysis）等）。

随着单位面积芯片中集成的晶体管数量的上升，进行这些分析测试的难度也在不断加大，以供电网络分析为例，随着芯片中节点数的不断提升（数千万以上），将芯片抽取成系统并求解的代价不断上升 (Ye 等, 2018)。因此，对高集成度的芯片的分析测试愈发困难。

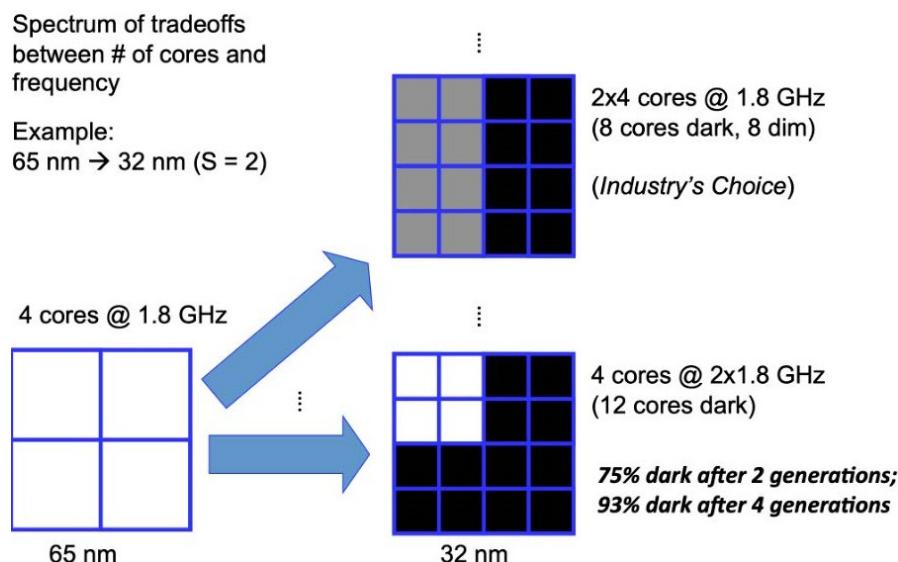


图 1.5 暗硅现象示意图 (Goulding 等, 2010)。

Figure 1.5 An example of dark silicon phenomenon (Goulding 等, 2010).

- 集成度上升带来的芯片功耗问题：由于登纳德缩放定律的失效，芯片上单位面积的功耗不断上升 (Esmaeilzadeh 等, 2011)，这导致了芯片上的热效应愈发严重。如图 1.5 所示，从 65nm 工艺到 32nm 工艺，缩放后的芯片相比此前具有

4倍的晶体管数量，频率工作在原来的2倍，由于利用率墙，芯片上将有75%的部分处于空闲的状态，也即暗硅（Dark silicon (Esmaeilzadeh等, 2011)）。目前工业界对暗硅的处理方法更倾向于让原有的2倍晶体管数量的电路工作在原有频率的低频状态，剩余的晶体管则处于闲置状态 (Taylor, 2012)。

1.3.2 阻变存储器的可靠性问题

阻变存储器的可靠性问题同样可分为两类：

- **写入型可靠性问题：**在对阻变存储器进行写入过程中，理想值和实际值之间的误差。常见的写入型可靠性问题包括：1) 固定型错误 (Stuck-at-fault, 简称 SAF (B. Zhang 等, 2019))，表示该阻变存储器发生了断路，或者无法形成导电丝。其造成的影响时该器件会被固定在高电导 (Stuck-at-1) 或者低电导 (Stuck-at-0) 状态，无法通过编程改变其电阻值；2) 变异型错误 (Variation (Yu 等, 2012))，表示阻变存储器器件在写入过程中，由于噪声（如写入电压的不稳定性）或器件电学参数的随机性导致的误差。
- **使用型可靠性问题：**在使用阻变存储器的过程中，因客观原因使得阻变存储器的电导值发生变化带来的误差。常见的使用型可靠性问题包括：1) 电导漂移错误 (Drift (Y. Ma 等, 2021a))，表示在使用的过程中，阻变存储器的电导值由于工作电压（相对于写入电压小很多）的作用，根据加压方向不断增大（减小）的问题；2) 热敏问题 (Thermal (C. Walczyk 等, 2011))，表示由于芯片温度而导致的阻变存储器电导值的变化。阻变存储器可使用的电导值范围将随着温度的上升而显著减少。

1.4 本文关注的问题

基于阻变存储器的芯片可以提高芯片集成度以及改进芯片计算架构，然而其在可靠性上仍有缺陷。此外，随着集成度的上升，对可靠性问题进行分析和改进的难度也不断加大。本文针对基于阻变存储器的加速器芯片的可靠性问题进行研究。本文研究的可靠性问题分为两类：1) 加速器电路上供电网络的可靠性；2) 加速器中阻变存储器交叉阵列的热敏可靠性。

在供电网络的可靠性问题中，目前存在的主要问题是供电网络的分析较为困难。随着电路规模的不断上升，对其进行精确分析的难度不断加大 (Xie 等,

2020)。近年来针对供电网络分析问题，前人主要从两个方面入手解决

1. 系统分析法：通过将芯片上的供电网络抽象(或近似)成线性系统，通过求解该线性系统来获得各个节点的电压值以及供电网络上的 IR drop 情况 (Kozhaya 等, 2002; Ye 等, 2018; Zhao 等, 2002)。然而，随着芯片上节点数的指数级增长，系统分析法在对线性系统进行求解的过程愈发困难。

2. 智能分析法：使用人工智能的方法，通过从芯片中抽取的各种参数（如模块坐标，切换状态等）以及系统分析得到的部分供电网络分析数据，训练出可对供电网络进行近似估计的模型 (Y C. Fang 等, 2018; Xie 等, 2020)。然而，在分析中需要的供电网络分析数据同样受到节点数增长的制约，此外，用这些方法训练出的模型不具有足够的稳定性，在不同的电路下往往需要重新训练。

在对芯片的供电网络分析的研究上，本文专注于系统分析法。由于目前的加速器芯片中存在多核同构的现象，如阻变存储器层的相同 ReRAM TILE 以及逻辑层的相同 GPU TILE (Joardar 等, 2019; Shafiee 等, 2016)，因此，本文针对多核同构的现象，尝试对供电网络的建模和求解的过程进行加速。

在阻变存储器交叉阵列的热敏可靠性问题中，该问题主要体现在其作为计算单元时产生的功耗较为集中，会在计算区域产生较高的温度，进而影响到计算单元的计算准确度。针对这一问题，前人主要从三个方面入手解决：

1. 降低权重法：通过调整神经网络训练过程中的损失函数，降低需要映射到交叉阵列中的权重总量，从而减少映射后交叉阵列中的电导总量 (G. L. Zhang 等, 2021; S. Zhang 等, 2019)。

2. 权重排布法：通过对权重映射方式的调整，达到均衡不同交叉阵列中的电导值以及避免将大权重映射到较热区域的作用，从而减少交叉阵列产生的峰值温度，避免计算准确度降低 (Beigi 等, 2018a; Shin 等, 2020)。

3. 在线补偿法：通过在计算过程中，实时地根据芯片温度对交叉阵列生成的结果进行补偿，以减少温度对结果的影响 (X. Liu 等, 2019; Shin 等, 2020)。

以上三种方法均对阻变存储器的热敏问题有一定作用，本文在前人研究的基础上，提出了布局优化方法并进一步优化了权重排布算法：

1. 芯片布局优化：针对前人未考虑的芯片布局问题，本文尝试对芯片的布局进行优化，以避免高功耗区域的聚集，进而减少高功耗区域对芯片温度的影响。

2. 权重排布优化：通过对交叉阵列可能的输入进行预估后，本文考虑输入

分布对权重的影响，提出新的权重重排布方法。此外，本文还对前人的排布方式进行优化，减少了排布过程中所不必要的约束项，扩展了权重排布的最优解空间。

1.5 本文的组织结构

基于阻变存储器的加速器芯片具有较好的人工智能应用前景，然而其存在的可靠性问题限制了该加速器的发展。因此，本文针对加速器芯片上的供电网络可靠性问题以及加速器中阻变存储器交叉阵列的可靠性问题进行了研究。

在第二章中，本文首先对神经网络的基本概念和阻变存储器进行介绍，然后介绍基于阻变存储器的神经网络加速器中阻变存储器可靠性问题以及供电网络可靠性问题。

在第三章中，本文针对加速器芯片电路上的供电网络可靠性进行分析。基于对目前供电网络分析趋势的观察，本文找到了前人提出的层级结构分析方法的缺点，并加以改进。此后，本文针对该加速器芯片多同构核的实际情况，提出了针对多同构核的供电网络分析加速算法。本文通过实验验证了所提出方法的有效性，并对实验结果进行了分析和讨论。

在第四章中，本文针对加速器芯片中交叉阵列上的热敏可靠性问题进行改进。根据对前人提出的权重排布方法的分析总结，本文首先从结构调整的方向考虑，通过调整加速器芯片的结构，使得芯片结构对功耗分布的影响降低，此后，本文通过考虑神经网络的输入分布以及对交叉阵列进行权重排布的限制，提出改进的权重排布方法以均衡交叉阵列上的功耗分布。实验结果证明了本文提出方法的有效性。

第五章对本文的工作进行总结，并在总结的基础上对未来进一步的工作进行展望。

第2章 基于阻变存储器的神经网络加速器芯片概述

本章对基于阻变存储器的神经网络加速器的基本概念进行介绍。

2.1 神经网络

神经网络最早在上个世纪四十年代由美国心理学家麦克洛奇 (McCulloch) 和数学家皮兹 (Pitts) 提出 (Cowan, 1990)。神经网络通过模拟动物大脑神经元的行为来完成各种任务 (如识别, 预测等)。图 2.2 展示了神经元的结构。由图可知, 单个神经元由树突 (Dendrite)、胞体 (Soma) 和轴突 (Axon) 组成。前后神经元之间通过突触 (Synapse) 将信号从前一个神经元的轴突传递到后一个神经元的树突上。

神经网络按照发展历程可被分为三种: 1) 多重感知机 (Multi-perceptron, 简称 MLP) (Hennessy 等, 2019); 2) 人工神经网络 (Artificial neural network, 简称 ANN) 以及深度神经网络 (Deep neural network, 简称 DNN) (Hinton 等, 2006); 3) 脉冲神经网络 (Spiking neural network, 简称 SNN) (Ghosh-Dastidar 等, 2009)。下文将对人工神经网络和脉冲神经网络进行介绍。

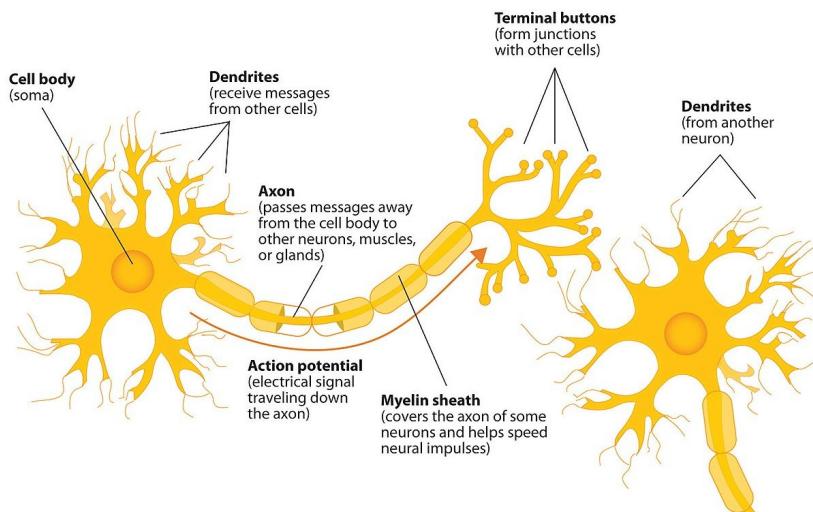


图 2.1 神经元结构 (Wikipedia, n.d.)。

Figure 2.1 Neuron structure (Wikipedia, n.d.).

2.1.1 人工神经网络

2.1.1.1 基础概念

人工神经网络主要由输入层、隐藏层、以及输出层组成，如图 2.2 所示。每一层之间通过突触（由权重矩阵 W 表示）连接，每一个神经元中包含激活函数（Activation function）和偏置（Bias）。公式 (2.1) 说明了前后神经元传递信息的过程：

$$\vec{y}_i = \phi \left(\sum_j \vec{x}_j \mathbf{w}_{i,j} + \vec{b}_i \right) \quad \dots (2.1)$$

其中 \vec{y}_i 表示第 i 层的输出， \vec{x}_j 是前一层（第 j 层）的输入， \vec{b}_i 表示第 i 层的偏置向量， $\mathbf{w}_{i,j}$ 表示第 i 层与第 j 层之间的权重矩阵。 ϕ 表示第 i 层的激活函数，常见的激活函数包括 ReLU (Agarap, 2018)，LeakyReLU (Xu 等, 2015)，以及 Sigmoid 等 (Mourigas-Alexandris 等, 2019)。他们的函数曲线如图 2.3 所示。

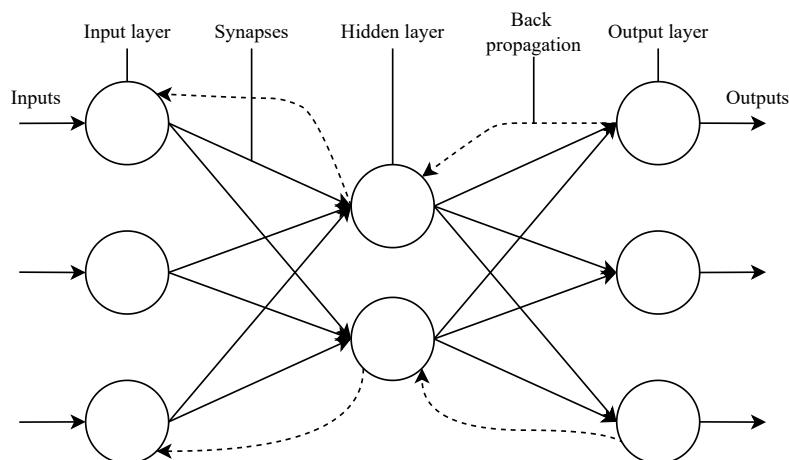


图 2.2 人工神经网络结构 (Abiodun 等, 2018)。

Figure 2.2 The structure of artificial neural network (Abiodun 等, 2018).

2.1.1.2 训练方法

人工神经网络主要通过反向传播（Back propagation，简称 BP）算法进行训练。其训练过程是基于数据的监督式学习，通过使用训练数据集（Training set）、

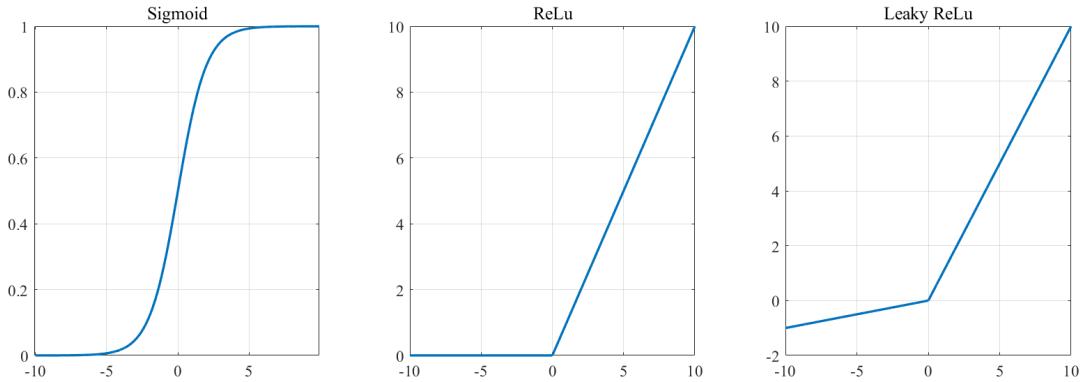


图 2.3 常见激活函数示例。

Figure 2.3 Examples of common used activation functions.

反向传播过程以及损失函数 (Loss function)，其从数据中学习并更新神经网络中的权重，其更新算法如公式 (2.1) 所示：

$$\begin{aligned} w_i &\leftarrow w_i - \frac{\partial L}{\partial w_i} \\ b_i &\leftarrow b_i - \frac{\partial L}{\partial b_i} \end{aligned} \quad \dots \quad (2.2)$$

其中 L 表示损失函数。在训练过程中，其将不断通过验证集 (Validation set) 验证当前模型的效果并根据结果调整网络的超参数 (Hyperparameter)，此外，在训练结束后，将在测试集 (Test set) 上测试最终的模型结果。

2.1.2 脉冲神经网络

脉冲神经网络是第三代神经网络，在经历了前两代 (MLP 和 ANN) 的发展后，脉冲神经网络在类脑化的方向上更进一步。不同于 ANN 中用连续值激活的神经元，SNN 采用了脉冲 (Spike) 激活的方式。这种方式也更类似与动物神经元的激活过程 (Tavanaei 等, 2019)。此外，SNN 相比于 ANN 具有低功耗的优势 (Y. Ma 等, 2021b)，更适用于边缘计算。

2.1.2.1 基础概念

SNN 的网络结构与 ANN 类似，其主要区别在于，1) 神经元之间传递的信号仅为离散值 (0 或者 1)；2) 引入了时间步 (Time step) 的概念，在每个时间步中，

神经元根据输入更新神经元的膜电位（Membrane potential，简称 MP）并根据当前神经元的激活阈值决定该神经元是否产生发射向下一层的脉冲。其主要使用的激活模型是累积-发射模型（Integrate-and-fire，简称 IF）（Vatajelu 等, 2019），该模型通过神经元上膜电位的累积以及阈值函数决定是否生成脉冲，如公式 (2.1) 所示：

$$z_i^l(t) = \sum_j w_{ij}^l \Theta_j^{l-1}(t) + b_i^l \quad \dots (2.3)$$

其中 Θ_j^{l-1} 表示的是第 1 层第 j 个脉冲输入， w_{ij}^l 是权重值， b_i^l 是第 1 层第 i 个神经元的偏置。

在每一个时间步中，神经元的膜电位都会根据公式 (2.1) 不断累加。当神经元的膜电位在第 k 个时间步中达到或超过阈值后，该神经元将在当前时间步中产生一个传递向后一层的脉冲信号，并将该神经元的膜电位重置为静息电位（Rest potential）。

在运行 SNN 之前，需要设定一个最大的时间步数量 T_{max} ，当运行的时间步达到 T_{max} 后，SNN 将输出最后的结果。

2.1.2.2 训练方法

脉冲神经网络的训练与人工神经网络不同。由于其脉冲的输入输出特性，公式 (2.1) 难以应用在 SNN 的训练中。目前学术界主流的对 SNN 的训练方法主要有两种：

1. 脉冲时序依赖可塑性方法(Spike-timing-dependent plasticity, 简称 STDP) (Lee 等, 2018)：该方法根据一个特定神经元的动作电位输入与输出的相对时间，去调整连接的强度。其权重调整方法如公式 (2.4) 所示：

$$\Delta w = \eta_{STDP} (e^{\frac{t_{pre}-t_{post}}{\tau_{pre}}} - \chi_{offset}) (w_{max} - w) (w - w_{min}) \quad \dots (2.4)$$

其中 Δw 表示对当前权重的改变量， η_{STDP} 表示学习率 (Learning rate)， $t_{pre} - t_{post}$ 表示前后激活脉冲的时间差， τ_{pre} 表示控制 STDP 时间窗长度的时间常量， χ_{offset} 表示阈值以及 w_{max}, w_{min} 表示权重的上下界。

2. ANN-SNN 转化 (Conversion) 方法 (Diehl 等, 2015)：将已训练好的 ANN 网络权重 (偏置) 转化成 SNN 所需的权重 (偏置)，以此生成所需的 SNN。通过

构建一个与 ANN 相同的网络结构以及公式 2.5，将 ANN 的权重归一化为合理的 SNN 的权重。其中 λ^l 表示第 l 层的归一化系数。

$$\begin{aligned} \mathbf{W}^l &= \mathbf{W}^l / \left(\frac{\lambda^l}{\lambda^{l-1}} \right) \\ \vec{b}^l &= \vec{b}^l / \lambda^l \end{aligned} \quad \dots (2.5)$$

2.2 基于阻变存储器的神经网络加速器芯片

基于阻变存储器的神经网络加速器芯片（下称加速器）可以通过基尔霍夫定律，有效的加速乘加运算。因此，其目前被学术界当作加速神经网络的理想路线进行研究。本节主要介绍阻变存储器的基础概念以及该加速器的计算架构。

2.2.1 阻变存储器

阻变存储器主要由三个结构组成，分别是顶层电极（Top electrode，简称 TE）、开关层（Switching layer）和底层电极（Bottom electrode，简称 BE）。图 2.9 展示了阻变存储器的结构以及复位（Reset）/置位（Set）过程。从图中可以看出，当阻变存储器处于低阻态时，TE 和 BE 通过连续的导电丝（Conductive filament，简称 CF）连在一起，在复位过程中，阻变存储器逐渐减少 TE 与 BE 之间的导电丝长度与宽度，直到 TE 与 BE 完全分离。此时阻变存储器达到高阻态。置位过程则与复位过程相反，通过不断延伸电导丝的长宽，连接 TE 与 BE，以达到将低电阻的目的（Ambrosi 等, 2019）。

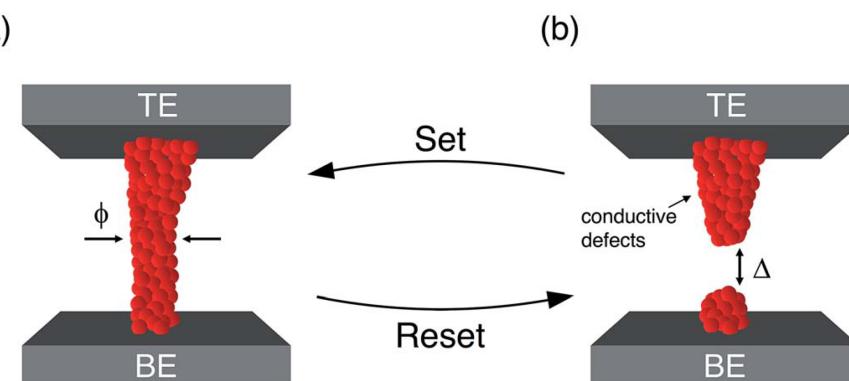


图 2.4 (a) 低阻态 (LRS) 下的阻变存储器。(b) 高阻态 (HRS) 下的阻变存储器。（Ambrosi 等, 2019）

Figure 2.4 (a) Low resistance state (LRS). (b) High resistance state (HRS). (Ambrosi 等, 2019)

不同材质下的阻变存储器的电流-电压曲线如图 2.5 所示：

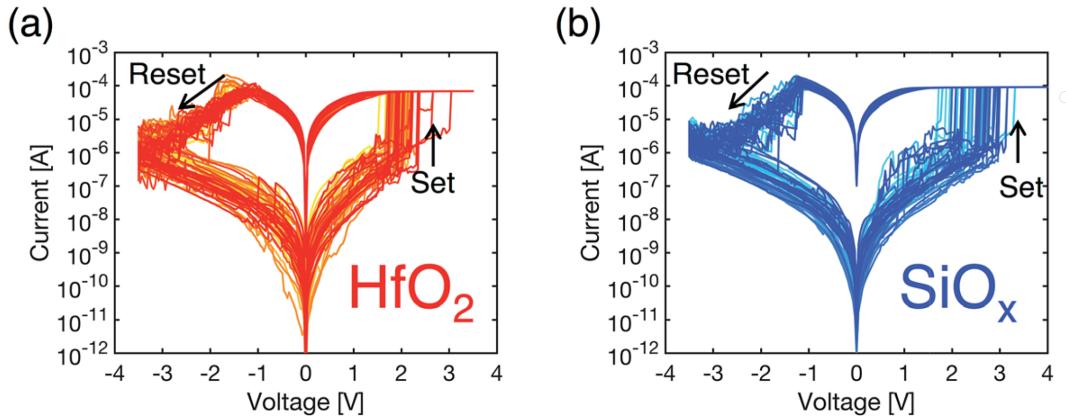


图 2.5 (a) 基于 HfO_2 器件的阻变存储器 I-V 曲线。(b) 基于 SiO_x 器件的阻变存储器 I-V 曲线。 (Ambrosi 等, 2019)

Figure 2.5 Measured I - V curves for (a) HfO_2 and (b) SiO_x , (Ambrosi 等, 2019)

2.2.2 神经网络加速器的芯片架构

本节将对基于阻变存储器的交叉阵列（下称交叉阵列）能加速向量-矩阵运算的原理以及基于阻变存储器的加速器芯片（下称加速器）架构进行介绍。

2.2.2.1 映射和计算方法

基于阻变存储器的交叉阵列根据其构成元件的不同，主要有三种不同的类型：1) 1R (One-memristor) (T. Li 等, 2017)、1S1R (One-selector-one-memristor) (J. J. Huang 等, 2011) 和 1T1R (One-transistor-one-memristor) (Z. Fang 等, 2013)。其中，由于 1T1R 结构可以有效的减少写入过程中 sneak path 的影响 (Zangeneh 等, 2012)，以及目前主流的计算架构使用的也是 1T1R 结构，因此，本节将以 1T1R 结构为例进行介绍。

图 2.6 中右图展示了基于 1T1R 的交叉阵列结构。从图中可以看出，在该结构中，每一个基本单元都由一个晶体管以及一个阻变存储器构成。对该基本单元进行写入的操作流程为：1) 通过字线 (Word line, 简称 WL) 选中需要进行写入操作的器件的所在列；2) 通过改变源线 (Source line, 简称 SL) 和位线 (Bit line, 简称 BL) 之间的电压，以对器件的两端施加不同的电压来改变其电阻状态，从而存储数据。当需要对该单元读取数据时，仅需通过字线选中器件的所在

列，然后在期间两端施加电压（相对写电压较小），通过读取流经的电流来获得存储的数据。

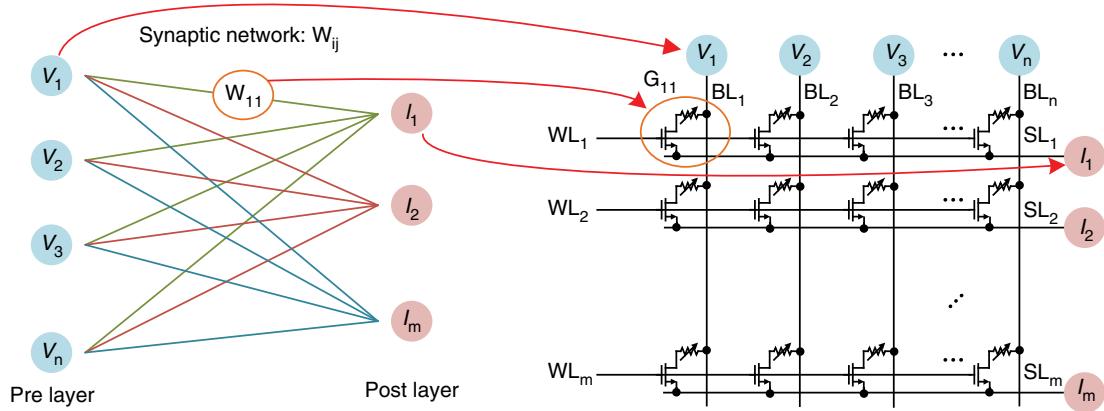


图 2.6 用于神经网络加速的 1T1R 交叉阵列 (Yao 等, 2017)。

Figure 2.6 1T1R crossbar structure for neural network accelerator (Yao 等, 2017).

在设置完交叉阵列后，另一个重要的操作是将向量与矩阵乘法中的输入向量与被积矩阵映射到阻变存储器上，图 2.6 以神经网络的映射为例展示了该映射过程。在执行乘加操作之前，加速器需要根据公式 2.6 (S. Zhang 等, 2019)，将矩阵元素（也即权重）映射到阻变存储器中。

$$G = \frac{G_{\max} - G_{\min}}{W_{\max} - W_{\min}}(W - W_{\min}) + G_{\min} \quad \dots (2.6)$$

其中 G_{\max} 和 G_{\min} 分别表示阻变存储器可表示的最大和最小的电导值， W_{\max} 和 W_{\min} 分别表示矩阵中最大和最小的权重值。在执行乘加操作中，加速器需要将交叉阵列的字线全部置位为“1”，并且把输入向量通过模数转换器转化为位线上的电压值，通过基尔霍夫定律 (Hu 等, 2016) 计算出向量-矩阵乘法的结果。

2.2.2.2 ISAAC 结构

对于基于阻变存储器的神经网络加速器芯片来说，目前学术界最流行的架构是 ISAAC 架构 (Shafiee 等, 2016)，其结构如图 2.7 所示：

该架构由多个近似的 TILE 组成，其中每一个 TILE 包含多个 IMA 单元 (In-situ multiply-accumulate units) 以及相应的缓冲存储器 (eDRAM buffer)、池化单元 (Max pool unit, 简称 MP) 等电路结构。而对于每一个 IMA 单元，其包含数

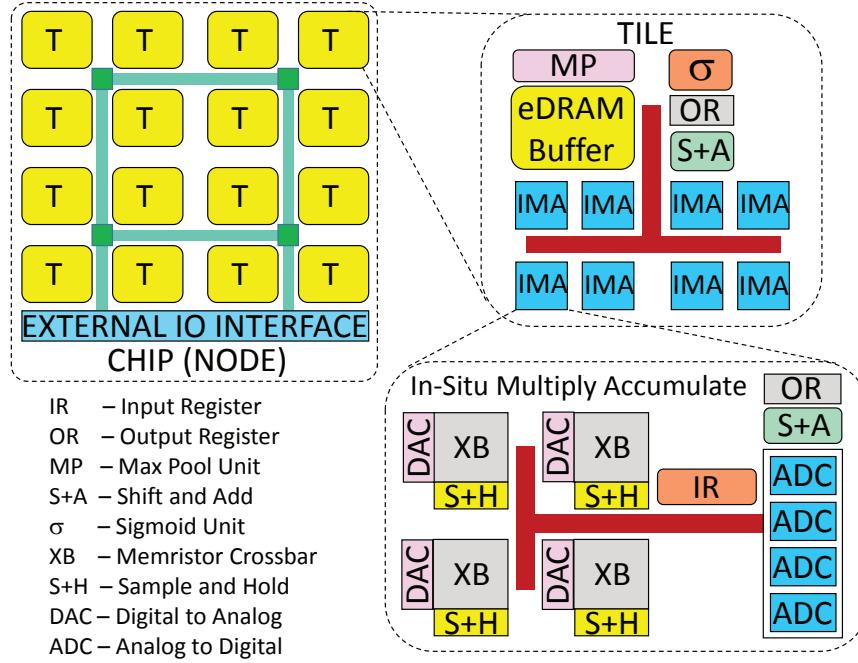


图 2.7 ISAAC 架构 (Shafiee 等, 2016)。

Figure 2.7 ISAAC structure (Shafiee 等, 2016).

一个基于阻变存储器的交叉阵列 (Crossbar, 简称 XB) 以及相应的辅助计算的电路结构, 如采样保持电路 (Sample and hold, 简称 S+H), 模数转换器 (Analog to digital converter, 简称 ADC) 等。

该加速器的工作流程如图 2.8 所示。其中图 2.8(a) 展示了卷积神经网络 (Convolutional neural network, 简称 CNN) 从第 i 层到第 $i+2$ 层的过程, 在其中, 输入图像经历了卷积和池化两个操作。对于该加速器来说, CNN 中的全部操作都可以在加速器中进行加速, 其加速流程如图 2.8(b) 中所示。ISAAC 采用流水线的结构处理数据, 其假定输入数据精度为 16 比特, 在接收到输入向量后, 其将输入向量按比特位分为 16 份, 在每一个时钟周期中, 其将 1 比特的输入向量经过输入寄存器 (Input register, 简称 IR), 数模转换器等电路结构, 将其转化为电压并输入到 IMA 中的交叉阵列中, 并使用采样保持电路存储下当前的计算结果, 该操作将重复 16 次。此后, 该结果将在下一个时钟周期被送入模数转换器中, 并在后续时钟周期中通过移位累加电路 (Shift and add, 简称 S+A), 输出寄存器 (Output register, 简称 OR) 以及激活函数计算出该层的计算结果。

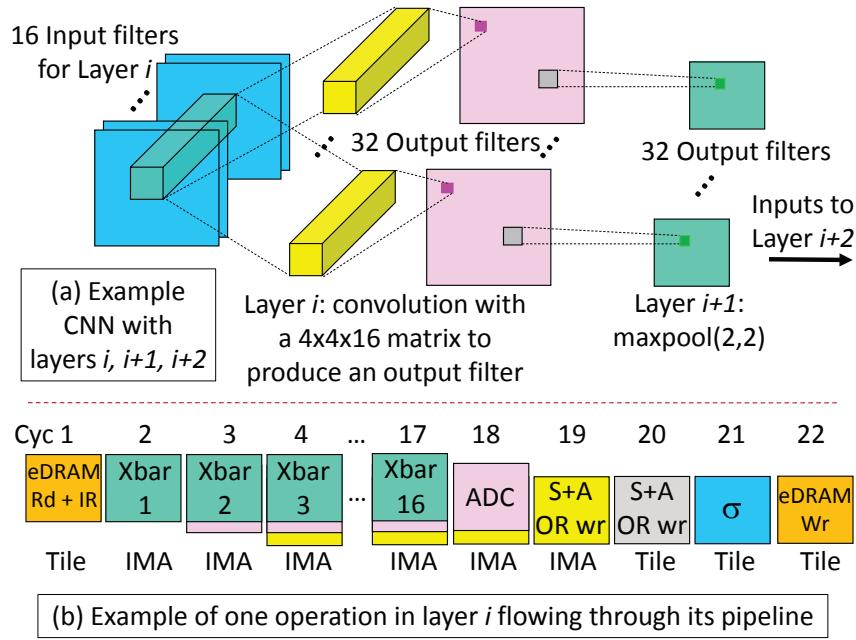


图 2.8 ISAAC 的工作流程图 (Shafiee 等, 2016)。

Figure 2.8 Workflow of ISAAC (Shafiee 等, 2016).

2.3 加速器芯片中阻变存储器的可靠性问题

2.3.1 可靠性问题概述

尽管阻变存储器具有低功耗, 高密度等特性, 然而其本身在可靠性上仍有不足。如小节 1.3.2 所述, 其可靠性问题主要分为两类:

- **写入型可靠性问题:** 在对阻变存储器进行写入过程中, 理想值和实际值之间的误差。常见的写入型可靠性问题包括: 1) 固定型错误 (Stuck-at-fault, 简称 SAF (B. Zhang 等, 2019)), 表示该阻变存储器发生了断路, 或者无法形成导电丝。其造成的影响时该器件会被固定在高电导 (Stuck-at-0) 或者低电导 (Stuck-at-1) 状态, 无法通过编程改变其电阻值; 2) 变异型错误 (Variation (Yu 等, 2012)), 表示阻变存储器器件在写入过程中, 由于噪声 (如写入电压的不稳定性) 或器件电学参数的随机性导致的误差。

- **使用型可靠性问题:** 在使用阻变存储器的过程中, 因客观原因使得阻变存储器的电导值发生变化带来的误差。常见的使用型可靠性问题包括: 1) 电导漂移错误 (Drift (Y. Ma 等, 2021a)) 表示在使用的过程中, 阻变存储器的电导值由于工作电压 (相对于写入电压小很多) 的作用, 根据加压方向不断增大 (减小) 的问题; 2) 热敏问题 (Thermal (C. Walczyk 等, 2011)) 表示由于芯片温度而导致

的阻变存储器电导值的变化，阻变存储器可使用的电导值范围将随着温度的上升而显著减少。

本文主要针对阻变存储器的热问题进行研究。

2.3.2 阻变存储器的热敏问题

在使用阻变存储器进行计算或者存储时，学术界和工业界均希望其存储的阻值较为稳定，不会轻易受到外界干扰。然而根据文献 (C. Walczyk 等, 2011)，阻变存储器的 I-V 曲线将随着温度的变化而变化。其变化曲线如图 2.9 所示：

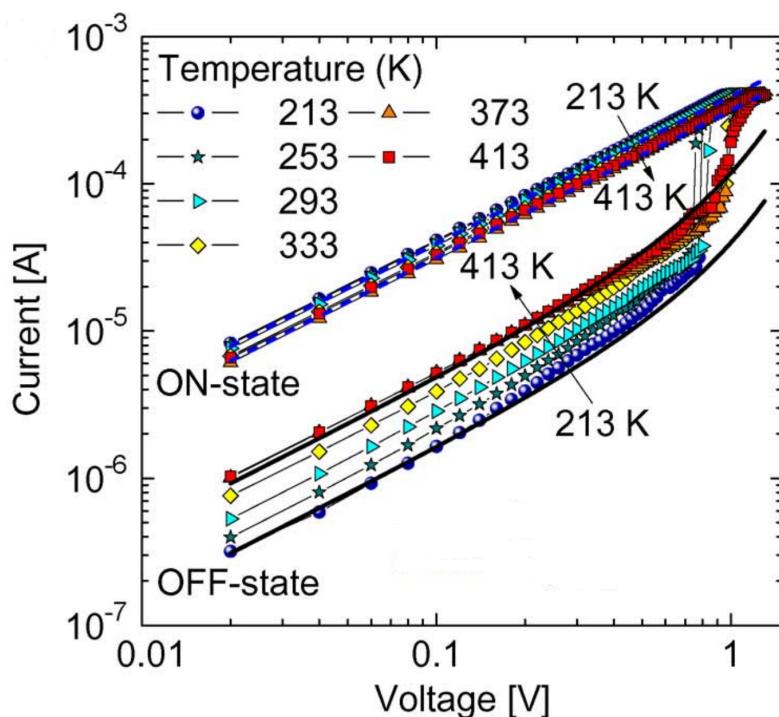


图 2.9 热效应用下阻变存储器的电流-电压曲线 (C. Walczyk 等, 2011)。

Figure 2.9 The I-V curve of ReRAM with thermal effect (C. Walczyk 等, 2011).

图中 ON-state 和 OFF-state 分别表示低阻态和高阻态下的阻变存储器。从图中可以看出，阻变存储器的阻值随温度的变化而变化，其变化趋势是：低阻态的电阻值将随温度升高而升高，高阻态的电阻值将随温度的升高而减少。

文献 (Beigi 等, 2018a) 展示了其得出的阻变存储器的电导值随温度的变化曲线，如图 2.10 所示。从图中可以看出，尽管与图 2.9 中高低阻态的变化幅度不同，但其也遵循了阻变存储器的阻值与温度的变化趋势。

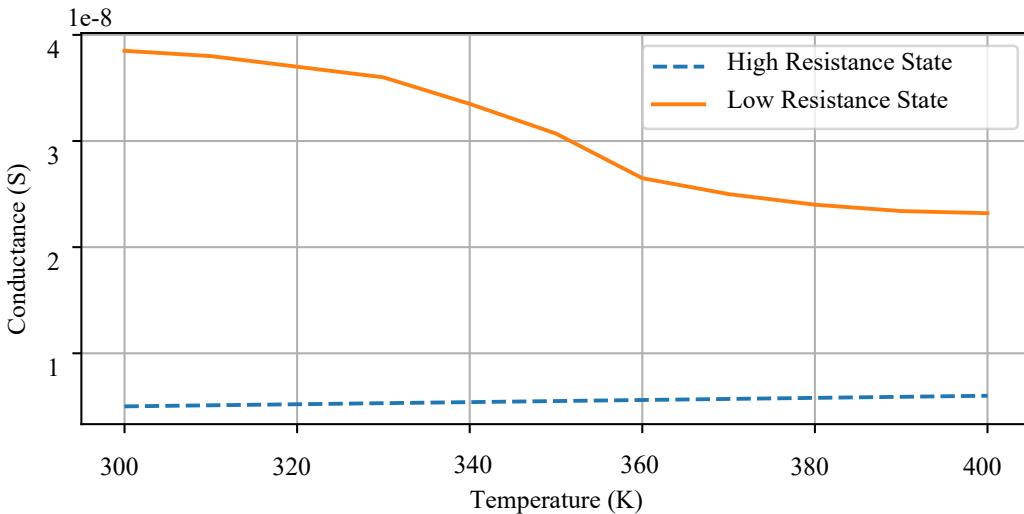


图 2.10 热效应作用下阻变存储器电导值的变化曲线 (Beigi 等, 2018a)。

Figure 2.10 The conductance variation of ReRAM with thermal effect (Beigi 等, 2018a).

2.4 加速器芯片中供电网络可靠性问题

2.4.1 供电网络概述

片上供电网络指的是对芯片进行供电的网络，其组成结构主要分为连接到电压源 VDD 的使能网络以及连接到 GND 的共地网络 (Nassif, 2008)。片上供电网络是高性能集成电路的重要组成部分 (Qian 等, 2004)，其可靠性分析结果将会决定当前供电的情况下芯片的工作性能，如功耗，时序等。由于近年来芯片集成度的上升给芯片带来了更多的电源噪声 (刘婷婷, 2011) 以及导线电阻增大 (Nithin 等, 2010) 等情况，因此，片上供电网络分析仍对芯片至关重要。

对于加速器芯片来说，根据不同的架构，其芯片可能会与传统的 CPU 或 GPU 进行结合 (Joardar 等, 2019)，且其本身的阻变存储器交叉阵列及其附属电路也具有较大的供电网络。以 ISAAC 结构为例，其中 1 个 TILE 包含 12 个 IMA，每一个 IMA 包含 8 个 XB，即使仅计算全部 XB 中的晶体管供电网络，该网络也具有约 152 万个节点。因此，对加速器上的供电网络分析方法改进也是必要的。

2.4.2 层级式供电网络分析方法

本节概述了层级式供电网络分析 (Hierarchical analysis, 简称 HA) (Zhao 等, 2002) 的一般情况。HA 方法的目的是克服传统压降分析中使用宏观模型的容量限制。该方法设计了一个层级式结构模型，将片上供电网络划分为全局供电网

络和局部供电网络。经过该方法建模后，每个局部供电网络的数学模型为：

$$\underbrace{\begin{bmatrix} G_{11} & G_{12} \\ G_{12}^T & G_{22} \end{bmatrix}}_G \underbrace{\begin{bmatrix} U_{int} \\ V_{port} \end{bmatrix}}_V = \underbrace{\begin{bmatrix} J_{int} \\ J_{port} + I \end{bmatrix}}_J \quad \dots (2.7)$$

在这里

- G_{11}, G_{22} : 分别表示内部节点之间的电导矩阵和端口节点之间的电导矩阵， G_{12} : 表示内部节点与端口节点之间的电导矩阵；
- U_{int}, V_{port} : 分别表示内部节点的电压向量和端口节点的电压源向量；
- J_{int}, J_{port} : 分别表示加载到内部节点和端口节点的电流源向量；
- I : 表示在接口处（端口节点处）流经的电流向量；
- G, V, J : 分别表示局部供电网络的点到矩阵，电压源向量以及电流源向量。

此处的等式 2.7 可转化为如下等式：

$$I = \underbrace{(G_{22} - G_{12}^T G_{11}^{-1} G_{12})}_\text{端口导纳矩阵 } A V_{port} + \underbrace{(G_{12}^T G_{11}^{-1} J_{int} - J_{port})}_S \quad \dots (2.8)$$

当全部的局部供电网络都通过该方法生成后，全局的供电网络即可通过 MNA 方法，以如下的形式表示出来：

$$\begin{bmatrix} G_{00} & G_{01} & G_{02} & \cdots & G_{0k} \\ G_{01}^T & A_1 & G_{12} & \cdots & G_{1k} \\ G_{02}^T & G_{12}^T & A_2 & \cdots & G_{2k} \\ \vdots & & & \ddots & \vdots \\ G_{0k}^T & G_{1k}^T & G_{2k}^T & \cdots & A_k \end{bmatrix} \begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ \vdots \\ V_k \end{bmatrix} = \begin{bmatrix} I_0 \\ -S_1 \\ -S_2 \\ \vdots \\ -S_k \end{bmatrix} \quad \dots (2.9)$$

在这里

- 全局节点索引将被标记为 0，
- I_0 : 表示流出全局节点的电流源向量；
- G_{ij} : 表示各个局部供电网络之间的电导矩阵，也即局部供电网络之间的联系；
- S_i, V_i, A_i : 分别表示由 2.8 生成的第 i 个局部供电网络的 S 常数，电压源向量以及端口矩阵。

根据公式 2.9, 可以求解出全局供电网络的解 (也即端口节点 V_{port} 的解)。如此, 局部供电网络的求解方案为:

$$U_{int} = G_{11}^{-1}(J_{int} - G_{12}V_{port}) \quad \dots (2.10)$$

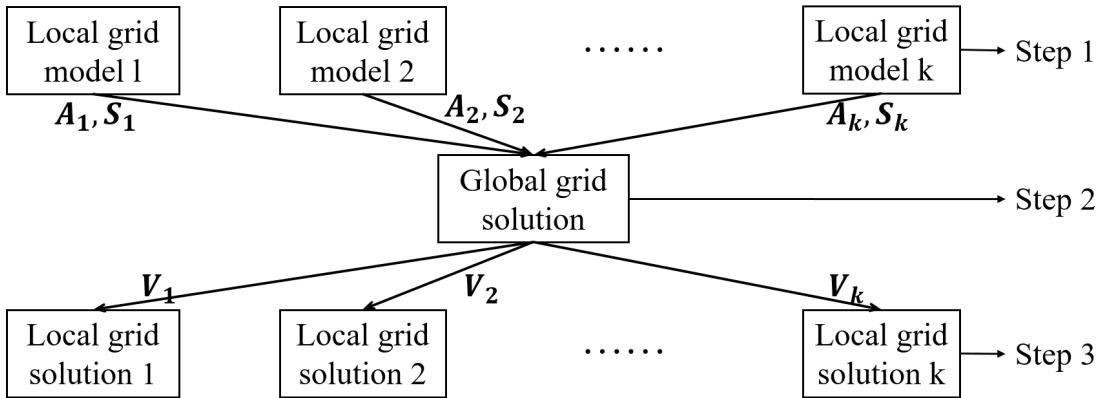


图 2.11 层级式供电网络分析流程 (Zhao 等, 2002)。

Figure 2.11 Flow of hierarchical analysis (Zhao 等, 2002).

根据上述方案, 可以得出 HA 方法通过以下步骤来求解整个供电网络, 全部步骤如图 2.11 所示:

- *Step 1*: 生成层级式模型, 根据公式 2.8 对全部的局部供电网络求解出 A_i, S_i ;
- *Step 2*: 利用 step 1 所生成的 A_i, S_i 得到全局供电网络的节点方程, 并通过公式 2.9 求解出全局供电网络的电压情况;
- *Step 3*: 在求解出全局供电网络的电压情况后, 根据端口节点的电压, 通过公式 2.10 求解出各个局部供电网络的内部节点电压, 以此来分析在全部芯片上的电压降。

通过该方法, 设计者可在仿真阶段得到理想情况下的芯片的片上供电网络分析情况, 且得到的结果是完全精准的。基于这些结果, 设计者将可以对芯片内部的工作状态等情况进行进一步判断, 并据此改进芯片电路结构。

2.5 本章小结

本章首先介绍了人工神经网络和脉冲神经网络的基础概念和训练方法, 然后介绍了基于阻变式存储器的神经网络加速器的基本概念。此后, 本章介绍了

基于阻变式存储器的神经网络加速器的可靠性问题，最后，本章介绍了对该神经网络加速器芯片中的片上供电网络分析方法。在接下来的两章中，本文将针对该供电网络分析方法和阻变存储器的可靠性问题进行研究。

第3章 针对加速器芯片电路的供电网络分析加速算法

如第二章所言，片上供电网络（Power distribution network，简称 PDN）分析对芯片的可靠性至关重要，其分析结果将会说明芯片能否在当前供电的情况下良好工作。然而，随着芯片集成度的不断上升，如何快速的对片上供电网络进行分析成为目前研究的难点。本文主要对基于阻变存储器的加速器芯片（下称加速器）的片上供电网络进行分析。

传统的片上压降分析基于改进后的节点分析（Modified Nodal Analysis，简称 MNA）(Ho 等, 1975) 方法，该方法将片上网络转换成一个线性系统，通过线性方程来获得芯片上各个节点的准确电压降，从而在设计阶段对存在较大电压降（产生较大功耗）的区域进行进一步的分析和改进。虽然这种传统的方法十分准确，但是随着节点数量的不断上升，该方法对计算能力和内存需求的开销也随之增加，求解该线性方程组也变的愈发困难。

此外，近年来研究者也提出许多基于人工智能（如机器学习或深度学习）来对 PDN 进行近似分析的方法，然而其结果往往不够精确或无法适用不同类型的电路。因此，为了获得精准而快速的 PDN 分析方法，本文针对传统的基于层级结构的供电网络分析方法（Hierarchical analysis，简称 HA）(Zhao 等, 2002)，提出了一种改进后的层级结构分析方法（Improved hierarchical analysis，简称 IHA）来对多核同构芯片进行精确分析。本文的主要研究背景是在静态工作条件下的加速器 PDN 分析，然而，由于缺乏合适的开源数据，本文仿照前人的方法 (Ye 等, 2018)，通过数学手段以及自构建的数据集对问题进行建模和验证。

在下文中，本文将首先介绍对现有加速器芯片的观察与分析，在此之后，本文将对 HA 方法进行改进，并针对多核同构芯片提出针对性的加速分析算法，最后通过实验验证改进后的方法对分析速度的提升。实验结果表明，本文提出的方法可以在 8 核同构的加速器芯片上达到 7.17 倍的分析加速以及 99.9% 的精度，且该加速倍数与同构核数成正比关系。而在近同构的 8 核加速器芯片上，本文的方法可达到最高 5.21 倍的分析加速以及 99.9% 的精度。

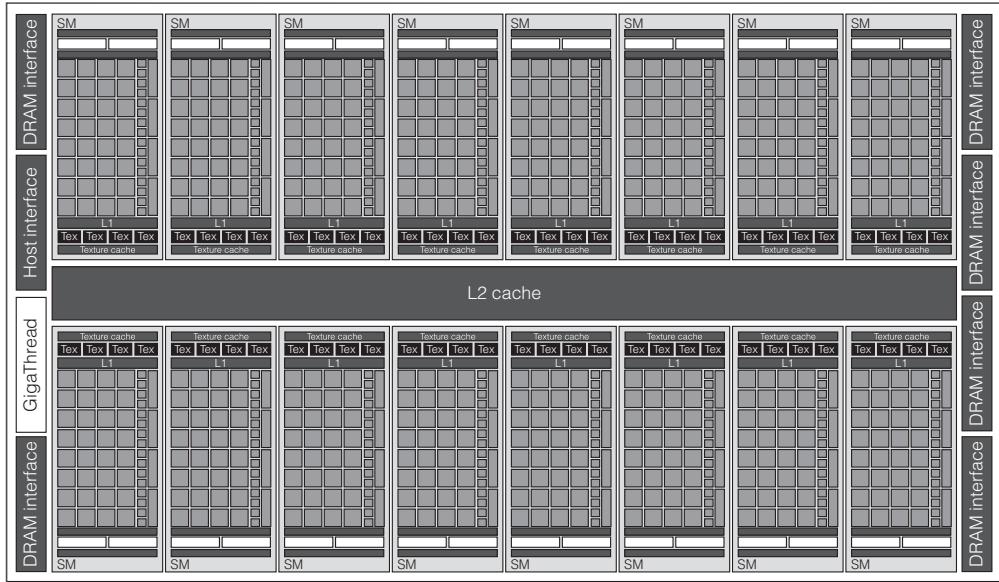


图 3.1 GPU 的计算架构 (Nickolls 等, 2010)。

Figure 3.1 GPU computing architecture (Nickolls 等, 2010).

3.1 观察与动机

在小节 2.4 中, 本文介绍了基于层级结构的供电网络分析方法, 该方法大大缩短了对供电网络分析的运行时间, 并提供了准确的求解方案。然而, 随着芯片的发展, 这种方法逐渐出现一些弊端:

1. 在使用 HA 方法进行分析时, 芯片上各个局部供电网络不应该包含电压源。而在实际的芯片中, 并无法保证通过该方法构建的局部网络是无源网络, 因此, 需要对该方法进行一些改进以适应有电压源分析。
2. 由于目前芯片上多核的发展趋势, 不论是传统的基于 CMOS 工艺的芯片, 还是基于新兴器件的芯片, 在它们的结构中均会出现大量的同构或者近同构的核, 如中央处理单元 (Central processing unit, 简称 CPU) 或图形处理单元 (Graph processing unit, 简称 GPU) 中的多核结构 (Gepner 等, 2006; Nickolls 等, 2010)。图 3.1 展示了 GPU 中的架构。从图中可以看出, 芯片上的很多模块是相同或类似的。

传统的供电网络分析方法并没有针对这种特性进行分析。而在当前的加速器芯片中, 其结构包含了大量同构核, 如阻变存储器层的相同 ReRAM TILE 以及逻辑层的相同 GPU TILE (Joardar 等, 2019; Shafiee 等, 2016)。这种情况提供了一种针对同构核进行 PDN 分析加速的机会。

本文将在小节 3.2 中阐述对 HA 方法不能对含电压源的网络进行分析的限制的移除方法，以及在小节 3.3 中提高 HA 方法在同构核上的效率。

3.2 对层级式供电网络分析方法的改进

本小节将 HA 方法扩展为可对存在电压源的供电网络进行分析的方法。考虑图 3.2 所示的局部电路，其中节点 1 包含一个电压源 V_{dd} ，节点 4 包含一个电流源 J 。节点 2 是从全局供电网络中接收外部电流 I 的端口节点。这里本文假设电压源和节点 1 之间的电阻为零。

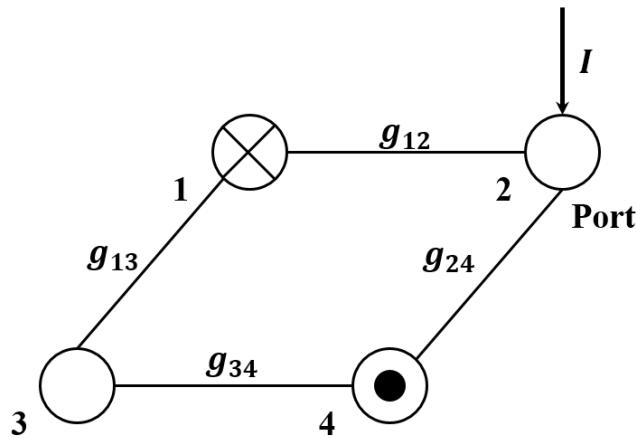


图 3.2 电路实例。节点 1 连接电压源，节点 4 连接电流源，I 表示外部电流。

Figure 3.2 Example circuit. Node 1 connects to a voltage source, Node 4 connects to a current source, I represents the external current.

首先，根据图 3.2 所示的电路可构建出该电路的节点方程：

$$\begin{bmatrix} G_1 & -g_{12} & -g_{13} & 0 & 1 \\ -g_{12} & G_2 & 0 & -g_{24} & 0 \\ -g_{13} & 0 & G_3 & -g_{34} & 0 \\ 0 & -g_{24} & -g_{34} & G_4 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ I_v \end{bmatrix} = \begin{bmatrix} 0 \\ I_{port} \\ 0 \\ J \\ V_{dd} \end{bmatrix} \quad \dots (3.1)$$

然后，基于 (Kozhaya 等, 2002) 的公式重构方法，通过消去与电压源直接相关的参数 V_1 和 I_v ，等式 (3.1) 可被转换为 (3.2)。该方法可将求解目标明确为未知的节点电压。

$$\begin{bmatrix} G_3 & -g_{34} & 0 \\ -g_{34} & G_4 & -g_{24} \\ 0 & -g_{24} & G_2 \end{bmatrix} \begin{bmatrix} V_3 \\ V_4 \\ V_2 \end{bmatrix} = \begin{bmatrix} 0 + g_{13}V_{dd} \\ J \\ I_{port} + g_{12}V_{dd} \end{bmatrix} \quad \dots (3.2)$$

因此，改进后的层级分析模型为：

$$\begin{bmatrix} G'_{11} & G'_{12} \\ G'^T_{12} & G'_{22} \end{bmatrix} \begin{bmatrix} V_{int} \\ V_{port} \end{bmatrix} = \begin{bmatrix} J_{int} + J_{int,v} \\ I_{port} + J_{port} + J_{port,v} \end{bmatrix} \quad \dots (3.3)$$

$$I_{port} = \underbrace{(G'_{22} - G'^T_{12} G'^{-1}_{11} G'_{12}) V_{port}}_{A'} + \underbrace{(G'^T_{12} G'^{-1}_{11} (J_{int} + J_{int,v}) - (J_{port} + J_{port,v}))}_{S'} \quad \dots (3.4)$$

其中 $J_{int,v}$ 和 $J_{port,v}$ 分别表示内部节点和端口节点的电压电流效应。通过该改进，HA 方法也可适用于含电压源的供电网络分析，同时，该方法也略微减小了电导矩阵的维数。

3.3 对同构芯片的加速分析方法

在处理同构芯片时，本文假设各核之间的拓扑结构和电阻值差异较小。同时，本文还假设在此处出现的全局供电网络是经过精心设计的，以此可以为每个同构核的供电网络提供需要的电压。

本文主要针对 HA 分析中的 Step 1 和 Step 3 部分进行加速。假设一个电路有 k 个独立的同构核，每个核包含 n 个节点。HA 方法会将这 k 个同构核通过公式(2.8)建模，其余部分作为全局的供电网络，通过同构核的建模结果及公式(2.9)进行建模。而在本文中，由于各个同构核之间具有相似性，所以，本文所使用的 IHA 方法首先在 Step 1 中用平均电导矩阵 \bar{G} 和平均当前资源向量 \bar{S}' 来替代公式(2.9)中的矩阵 A, S 。 \bar{G} 和 \bar{S}' 将通过公式(3.5)生成：

$$\bar{G} = \frac{1}{k} \sum_{i=1}^k G_i, \quad \bar{S}' = \frac{1}{k} \sum_{i=1}^k S'_i \quad \dots (3.5)$$

在此之后， \bar{G}, \bar{S}' 将会被带入至公式(2.9)去替换所有的 $A_j, S_j, j \in \{1, 2 \dots k\}$ ，通过该近似方法，可以避免对同构核进行重复计算 A_j, S_j 。理论上来说，这种方法可以在 Step 1 的建模中节省大约 k 倍的时间，并能同步的减少对内存的占用量。

在完成对第一步的优化之后，对全局供电网络的分析可以在 *Step 2* 中，通过公式 (2.9) 分析出全局网络中节点电压 $V_i, i \in \{0, 1, \dots, k\}$ 的近似解。该近似解的精确程度将会受到同构核之间的差异度的影响，差异度越高，对全局供电网络的求解就会越不精确。为了减少该差异度对结果的影响，本文还将对 *Step 3* 进行改进。

在 *Step 3* 中，首先根据公式 (2.7)、(2.9) 和 (3.5) 获得 $\bar{V}, \bar{G}_{11}^{-1}$ 和 \bar{J}_{int} 。其次，通过公式 (2.10) 计算得出 \bar{U}_{int} 作为接下来进行的迭代求解的初始解，由于各个核之间是同构的，生成的表征也会十分接近，所以这份初始解将会比随机生成的初始解更加接近真实解。然后，本文使用一种基于 krylov 子空间的迭代算法—广义最小残差法（Generalized minimal Residual Method, GMRES）(Saad, 1993) 去求解出所有局部供电网络的精确解。此时需要注意的是，在计算精确解的过程中，局部供电网络在公式 (2.10) 的参数都将使用各个局部网络的真实值。GMRES 的求解步骤如图 3.3 所示，其中 A, x_0, b, ε 分别表示线性系统矩阵，初始解，节点方程右侧以及误差阈值， g_k 表示当前残差。

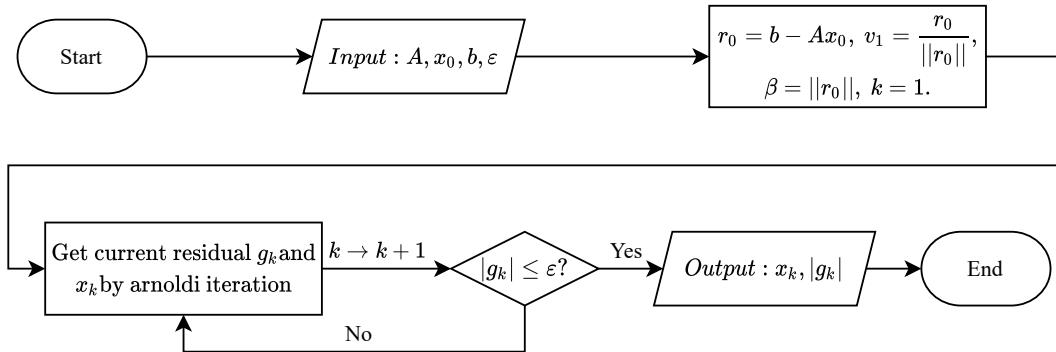


图 3.3 GMRES 算法流程图 (X. Ma, 2013)。

Figure 3.3 Flow chart of GMRES algorithm (X. Ma, 2013).

图中 Arnoldi 算法通过正交化生成 Krylov 子空间的标准正交基，并使用 QR 分解用来求解上 Hessenberg 矩阵的最小平方问题，进而实现对余量的最小化。

3.4 实验结果

3.4.1 实验设置

由于加速器芯片中同构的部分可能并不是完全类似，本文将同构核分为三种情况，以适应非完全同构（近同构）的情况：1) I型同构核，芯片内部的同构核在结构和阻值等参数上完全相同；2) II型同构核，芯片内部的同构核在结构上完全相同，但在阻值等参数上略有差异；3) III型同构核，芯片内部的同构核在阻值等参数上完全相同，但在结构上略有差异。

本文实验的数据集来源于美国国际商用机器公司 (International Business Machine, 简称 IBM) 和清华大学公开的供电网络数据集 (Nassif, 2008)，由于其数据并不包含多核同构的数据，我们在 IBM 公开的数据集上设计了一个新的具有同构结构的数据集。该数据集包含 4 个同构多核芯片 (textit{M1—M4})，分别包含 2, 4, 6, 8 个 IBM 数据集中 ibmpg1 芯片的 *Vdd_net* 部分。每个部分包含 12K 个节点，供电 *VDD* 为 1.8V，芯片内仅包含电阻，硅通孔，电压源以及电流源。限于数据集，本文仅考虑静态的供电网络分析。本文所有的实验都是在 Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz 和 8.00GB RAM 上进行的。实验结果如 3.4.2 小节所示。

3.4.2 实验结果分析

3.4.2.1 在 I 型同构芯片上的分析结果

在这一部分中，本文将分别使用 HA 和本文提出的 IHA 方法来分析这些数据集（即 I 型同构芯片）。其分析结果如表 3.1 所示，*Sp_all* 表示 IHA 与 HA 相比的运行速度加快程度。*Sp_1, 3* 表示 Step 1,3 中 IHA 比 HA 运行时加速程度。它们的计算方法如公式 (3.6) 所示：

$$Sp_{1, 3} = \frac{\text{Runtime in Step 1, 3 with HA}}{\text{Runtime in Step 1, 3 with IHA}} \quad \dots (3.6)$$

除此之外，表中第四列和第五列分别表示平均绝对值误差 (Mean Absolute Error, MAE) 和最大误差 (Max Error, MaxE)，在公式 (3.7) 和公式 (3.8) 中定义。其中参数 N 表示节点的个数，参数 \hat{y}_i, y_i 分别表示 IHA 方法和 HSpice 仿真结果在节点 i 处的电压值。

表 3.1 IHA 的分析在 I 型同构芯片上的分析结果。

Table 3.1 IHA in type I homogeneous chips.

Benchmark	Sp_all	Sp_I	Sp_3	MaxE(mV)	MAE(mV)
M1	2.04X	2.08X	1.83X	0	0
M2	4.56X	4.75X	3.66X	0	0
M3	5.88X	6.02X	5.88X	0	0
M4	7.17X	7.36X	7.17X	0	0

$$MAE = \frac{\sum_{i=1}^N \|\hat{y}_i - y_i\|}{N} \quad \dots (3.7)$$

$$MaxE = \max(\|\hat{y}_i - y_i\|), i = 1 \text{ to } N \quad \dots (3.8)$$

从表 3.1 中可得出，相对于 HA 方法，IHA 方法可以在不损失精度的情况下，在完全同构的芯片上达到最多 7.17 倍的供电网络分析加速。此外，可以看出随着核数的不断增长（从 2 核到 8 核），本文所提出方法的加速倍数也在不断上升（从 2.04 倍至 7.17 倍），因此可以得出，本文提出的方法在更多核上也同样具有良好的加速效果，且方法效果与供电网络的量级关系较小。

3.4.2.2 在 II, III 型同构芯片上的分析结果

考虑到各个同构核之间可能存在非完全一致的情况，比如考虑生产时的误差，或根据功耗需求所做出的一些局部调整等。本小节将对非完全同构（近同构）的芯片进行分析实验。

在本文的实验中，II 型同构芯片的生成方式为，在 I 型同构芯片的基础上不改变结构，同时，随机的选取每一个核中 10% 的电阻，将这些电阻以随机扰动的形式把这些电阻值乘以 $1 + \alpha$ ，其中 α 表示一个在 $[-5\%, 5\%]$ 内的均匀分布的一个随机采样点。如此，II 型同构核便有了同样结构和相似的电阻值。

对于 III 型同构核来说，其生成方式为随机挑选各个核中 0.1% 的电阻，将其阻值变为 1000Ω 。由于在 IBM 的数据集中电阻的阻值均在 1Ω 以下，所以这种改动将会使得被改动地方的电路形成近似开路的情况，从而达到改变电路结构的目的。如此，III 型同构核便有了相似结构核同样的电阻值。

表 3.2 IHA 的分析在 II 型同构芯片上的分析结果。**Table 3.2 IHA in type II homogeneous chips.**

<i>Benchmark</i>	<i>Sp_all</i>	<i>Sp_I</i>	<i>Sp_3</i>	<i>MaxE(mV)</i>	<i>MAE(mV)</i>
M1	1.93X	2.10X	1.26X	0.658	7.12e-2
M2	3.30X	3.87X	1.82X	0.935	1.18e-1
M3	4.64X	6.04X	2.03X	1.019	1.16e-1
M4	5.21X	7.00X	2.25X	1.184	1.09e-1

表 3.2 和 3.3 对 II, III 型同构核在 IHA 方法上的分析情况进行了展示。从表 3.2 可以看出，在这种情况下，总体的加速度相对 I 型同构核有明显的降低，该现象主要由 *Step 3* 导致，其原因是由于 II 型同构核之间的差异，使用迭代算法的时间不断上升。在精度上，可以看出 IHA 仍然能和 HSpice 的实际仿真结果具有极高的相似度，在 1.8V 的供电电压下仅有 1mV 的差异，精确度高达 99.9%。如果追求更高的精确度的话，可以对迭代算法的收敛条件进行进一步的约束，但此举也必然会带来计算时间的增加。

表 3.3 IHA 的分析在 III 型同构芯片上的分析结果。**Table 3.3 IHA in type III homogeneous chips.**

<i>Benchmark</i>	<i>Sp_all</i>	<i>Sp_I</i>	<i>Sp_3</i>	<i>MaxE(mV)</i>	<i>MAE(mV)</i>
M1	1.62X	1.97X	0.82X	1.82	8.9e-2
M2	2.60X	3.85X	0.94X	1.59	9.1e-2
M3	3.73X	5.92X	1.24X	1.56	7.8e-2
M4	4.03X	7.64X	1.12X	1.56	6.8e-2

表 3.3 展示了 III 型同构核在 IHA 方法上的分析情况，在 3 型同构核中，IHA 可以达到最多 4 倍的总时间加速，同时，也能保证分析结果具有足够的精度（99.9%）。

图 3.4 直观地展示了加速效果的比对图，从图中可以看出，对于全程的加速效果来说，在核的数量呈等差数列增长时，三种方法达到的加速效果都随着核数的增长而增长，而这其中以 type I 型的同构核增长效果最好，其加速效果几乎与

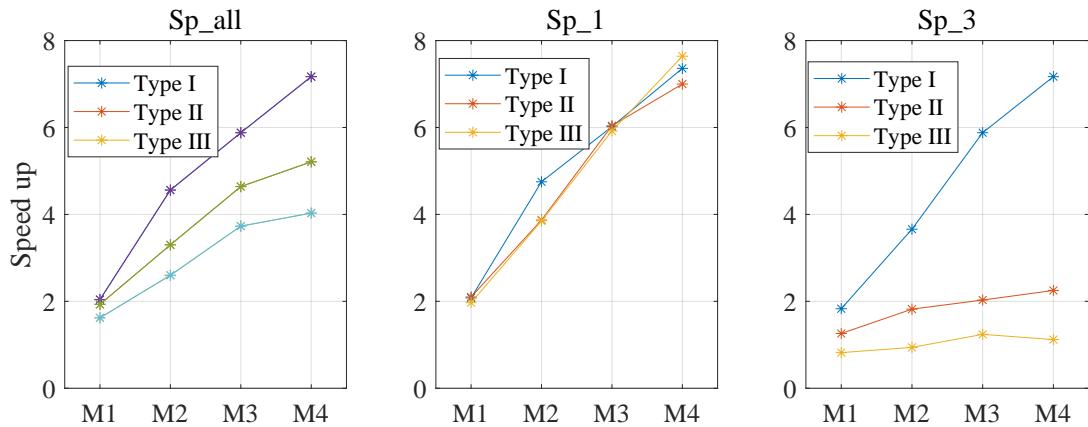


图 3.4 加速结果展示图。

Figure 3.4 Figure of the speedup results.

核数成正比关系。而对于 type II 以及 type III 型同构核来说，他们的效果仅在第一阶段 (Sp_1) 具有良好的加速效果，而在第三阶段 (Sp_3) 中，它们的加速效果均显著低于 type I 型同构核，其中以 type III 型同构核的效果最差，在核数较少的情况下甚至低于正常分析所需的时间。

本文认为这是由两个原因引起的：第一个原因是每个核之间的差值导致迭代算法收敛速度较慢。同时，比较表 3.2 和 3.3 可以看出，结构变化的影响大于阻值变化的影响；第二个原因是局部网格的大小，由于当前电脑仅能操作较小数据（数十万级别），迭代 GMRES 算法相对直接求解法没有足够的优越性。

在误差分析上，从表中可以看出，对于这三种同构核类型，本文提出的方法均具有较小的最大误差以及平均误差，其中以 I 型同构核效果最好，其误差均为 0。而对于其他两种类型的同构核，由于其最大误差也远小于供电电压 (1V)，因此，这两种方法的精度也可以得到保障。

3.5 本章小结

针对当今加速器芯片所具有的同构核特性，本文提出了一种基于同构核的 PDN 分析加速算法，加速了在多核同构芯片上 PDN 的分析速度。此外，本文还改进了基于层级结构的供电网络分析法，使之适用于片上含电压源的情况。实验表明，在 8 核同构的加速器芯片上，该方法在整体分析中可实现最高 7.17 倍的加速，且得出的电压结果与仿真结果几乎无差异，此外，由于该加速的倍数与

同构核数成正比的关系，因此本文实现的方法可以推广至更多核。在近同构的 8 核芯片上，该方法也能取得最高 5.21 倍的加速，精度为 99.9%。且根据实验结果，本文认为在近同构核的分析上，核的结构的改变相对核内参数值的改变对本文节提出的方法影响更大，很小的结构改变就会带来较大的时间消耗。

第4章 针对加速器芯片中交叉阵列的热敏可靠性研究

随着芯片集成度的提升，芯片的高功耗密度给基于 CMOS 的电路带来了暗硅等问题。神经网络加速器中基于阻变存储器的交叉阵列（下称交叉阵列）也会受到高功耗密度导致的热问题的影响。在温度升高时，在低阻态下的阻变存储器将展现出金属导体的特性，其电阻值会随着温度的上升而显著上升，而高阻态下的阻变存储器则展现出半导体的特性，其电阻值会随温度上升而略微下降 (C. Walczyk 等, 2011)。前人的实验结果说明，当阻变存储器的温度从 300K 上升至 400K 时，其低阻态的电阻值将上升近 1 倍，其高阻态与低阻态的差异将减少至原来的一半左右 (C. Walczyk 等, 2011)。这种变化使得阻变存储器在进行存储和计算中的精确度大幅下降 (Beigi 等, 2018a)，此外，该问题还会对阻变存储器的使用寿命造成极大影响 (Beigi 等, 2018b)。

前人解决该问题的方法主要有三种：1) 通过降低权重法，降低映射到交叉阵列中的总权重使得产生的总功耗降低 (G. L. Zhang 等, 2021; S. Zhang 等, 2019); 2) 通过权重排布法，均衡各个交叉阵列间的功耗，以降低交叉阵列产生的最大功耗密度 (Beigi 等, 2018a; Shin 等, 2020); 3) 在线补偿法，通过在线补偿，降低因温度带来的阻变存储器阻值变化对计算结果的影响。

本文针对权重排布法进行优化。在下文中，本文将首先说明对解决该问题的一些观察，并梳理本文的解决思路。接下来，本文将从三个方面对权重排布进行优化：1) 通过对加速器芯片的结构调整，使得芯片结构对功耗密度的影响降低；2) 通过对输入分布的考虑，对需要映射的神经网络权重进行加权以更适合后续权重排布；3) 通过去除前人所加的对权重排布的限制，扩充权重排布的最优解空间，并结合加权后的权重获得优化后的排布结果。最后，本文将对所提出方法进行实验及分析。实验结果表明，本文所提出的方法可在 VGG11 和 VGG9 网络上分别降低 5.0K 和 10.4K 的加速器芯片峰值温度，以及分别延长 1.3 倍和 1.72 倍的加速器使用寿命。

4.1 动机与解决思路

如前文所述，针对加速器芯片的热敏可靠性问题，现有的解决方案包括降低权重法、权重排布法以及在线补偿法三种。本文主要针对权重排布法进行优化改进，通过以下三个角度，阐述了本文提出方法的动机和解决思路：

- 加速器芯片的布局：由于芯片上的布局对片上的功耗分布具有很大影响，且目前的加速器芯片的布局并没有将对功耗的分析纳入考虑，其采用的结构汇集了大量高功耗密度的设备，因此会造成局部的功耗过高，并进一步对加速器芯片的性能造成影响，所以，本文针对加速器芯片的布局进行了改良，通过分离高功耗密度的设备以进一步均衡芯片上的功耗。
- 输入分布对功耗的影响：由于此前提出的权重排布法都只考虑了权重本身对功耗所带来的影响，而经过观察，输入分布也会影响加速器芯片中交叉阵列的功耗，进而影响芯片热分布和使用寿命。因此，本文提出了一种考虑输入的权重加权方法，以提升权重映射方法的效果。
- 映射方法：以往的映射方法没有考虑加速器中交叉阵列之间的行或列交换，而是在一个阵列内进行交换，或将所有阵列作为一个整体来处理，这样的处理方法会对他们提出的映射方法的最优解空间添加额外的限制，进而会影响映射方法的性能。其原因是由于实际上的加速器芯片往往包含多个相同或类似构架的阵列，如神经网络的某一层权重可能由多个阵列组成，而同一层的权重实际是可以进行相互交换的，也即阵列间的相互交换。基于此，本文提出了一种对行列同时进行交换的映射方法，来提高映射的效果。

4.1.1 加速器芯片的布局

本文的布局优化工作主要通过分散高功率密度组件来降低每个 TILE 的温度，通过对每一个 TILE 的优化，来达到对整体芯片的功耗均衡的效果。其动机来自于前人对加速器芯片的功耗及热敏可靠性相关的研究 (Beigi 等, 2018a; X. Liu 等, 2019; Shin 等, 2020)。在前人的研究中可知，芯片的布局对功率与热分析起着非常重要的作用，而目前的加速器芯片的架构并未考虑热问题。如果将热问题纳入加速器芯片的制作考量中，那么目前对基于阻变存储器的加速器芯片所采用的 ISAAC 结构并不合适。ISAAC 的结构组成在小节 2.2.2.2 中已介绍，其结构如图 2.7 所示。该结构在存内乘加单元 (In-Situ Multiply Accumulate, 简称

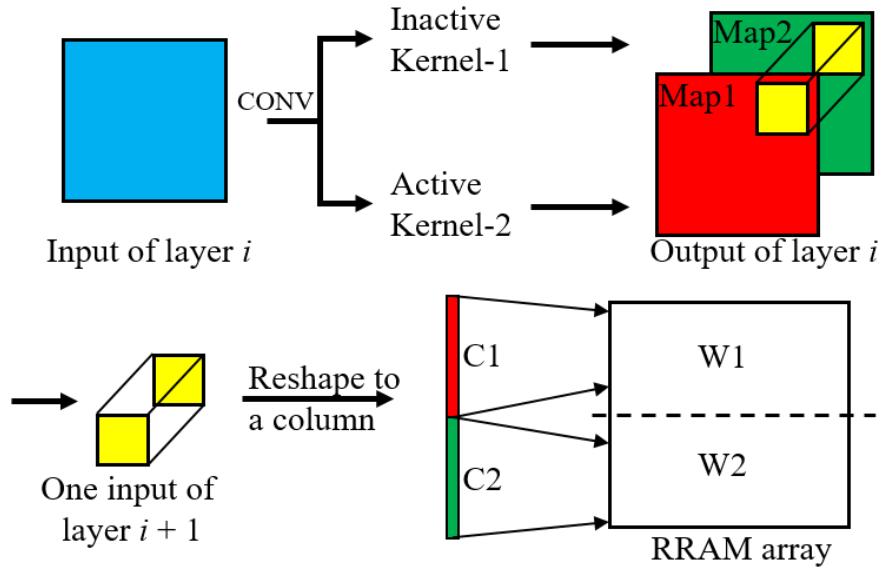


图 4.1 脉冲神经网络从第 i 层到第 $i + 1$ 层的推断过程。

Figure 4.1 One inference step in SNN from layer i to layer $i + 1$.

IMA) 内集中了交叉阵列和数模转换器，而这两者也是该结构中功率密度最高的两个元件，这种集中式的布局方式会比分散布置的条件下产生更高的温度，从而加剧热问题。因此，本文针对布局进行了优化，通过分离这些高功率密度的元件来更好的对功耗进行分布优化。

4.1.2 输入分布对功耗的影响

如小节 2.2.2.1 所示，交叉阵列是依据基尔霍夫定律，通过给阻变存储器两端加电压，根据流经电流的加和的方式来实现乘加运算，因此，在加速器芯片中，交叉阵列的功耗计算如公式 (4.1) 所示：

$$P_i = V_i^2 * G_i \propto Input_i^2 * Weight_i \quad \dots (4.1)$$

其中， P_i , V_i 和 G_i 分别为第 i 层交叉阵列的功率、输入电压和电导矩阵。 $Input_i$ 和 $Weight_i$ 分别表示第 i 层的输入和权值矩阵。

因此，在功耗的计算过程中，权重和输入分布均会对阵列产生的功耗产生影响，而由于前人对功耗分布的优化方法都集中在对权值进行调整，如减少整体的权值，对权重进行合理的映射，或是对权值进行补偿等方法，本文提出了针对输入分布进行映射的方法，其动机如图 4.1 所示：

该图以脉冲神经网络为例，展示了从第 i 层到第 $i + 1$ 层的推断过程。此处

Map_1 和 Map_2 表示第 i 层的输出结果，也可称为特征图（Feature map）。 C_1 和 C_2 分别表示进入第 $i+1$ 层中的一个卷积核（Convolutional kernel）的输入向量。 C_1 来自第 i 层的输出特征图 MAP_1 ，而 C_2 来自 i 层的另一个输出特征图 MAP_2 。 W_1 和 W_2 分别表示对应卷积核的交叉阵列。

由于不同的卷积核提取的特征不同，所以这些 kernel 的输出结果也不同。这里，我们假设第 i 层只有两个卷积核，不活跃的 Kernel-1 和活跃的 Kernel-2。不活跃的 Kernel-1 将会比活跃的 Kernel-2 产生更少的输出峰值。这样就会导致在第 $i+1$ 层中， C_1 会比 C_2 产生更少的激活值。如此，由于神经网络层层传递的顺序结构，这种情况会传递到下一层。假设权重矩阵 W_1 等于 W_2 ，那么根据公式 (4.1)，映射到 W_1 的交叉阵列比映射到 W_2 的消耗更少的功率。因此，神经网络的输入分布对交叉阵列产生的功耗具有一定影响，映射权重时需要考虑这一点。

对于这个推断，一个直观的解释是，假设该加速器芯片应用在人脸识别的任务上，那么对于该芯片来说，接收到的图片大概率会是类似人脸的图（差异过大的图会在前期的处理中被滤过或被处理 (Fan 等, 2011)），那么，假设该芯片所应用的网络在第一层包含两个卷积核，一个检测头发，另一个检测鼻子，那么可以很直观的理解到，检测头发的卷积核生成的特征图将会比检测鼻子的包含更多的脉冲。如果这两个卷积核后续的网络一样的话，检测头发的卷积核后续产生的功耗将比检测鼻子的要多。由此可见，交叉阵列的功率不仅仅取决于权值，还取决于一般情况下输入的分布。其功率、输入和权值之间的关系如式 4.1 所示。

因此，神经网络的输入分布对阵列产生的功耗具有一定影响，映射权重时需要考虑这一点。假设应用的网络有 n 层，其中第 i^{th} 层的输入脉冲可以用一个向量 $K_i, i = 1, 2, \dots, n$ 来表示。它们的值是通过将所有训练集输入训练良好的模型，并计算每一层的输入峰值来收集的。在这里，本文假设由超过 50,000 张的图像组成的训练集可以表示一般的输入情况（或者对于特定场景下的加速器芯片，特定的创建一个输入数据集或输入分布来表示一般的输入情况）。

图 4.2 展示了 VGG11 网络的第二层全连接层在训练集和测试集上归一化后的输入分布。该图的 x 轴为输入的一维向量，由于原本的输入是多维向量，较难以展示，因此这里本文将输入向量重构为一维向量来更直观地表示这一层的输入分布。y 轴是这一层接收到的脉冲的归一化数量。从图中我们可以看出，输入

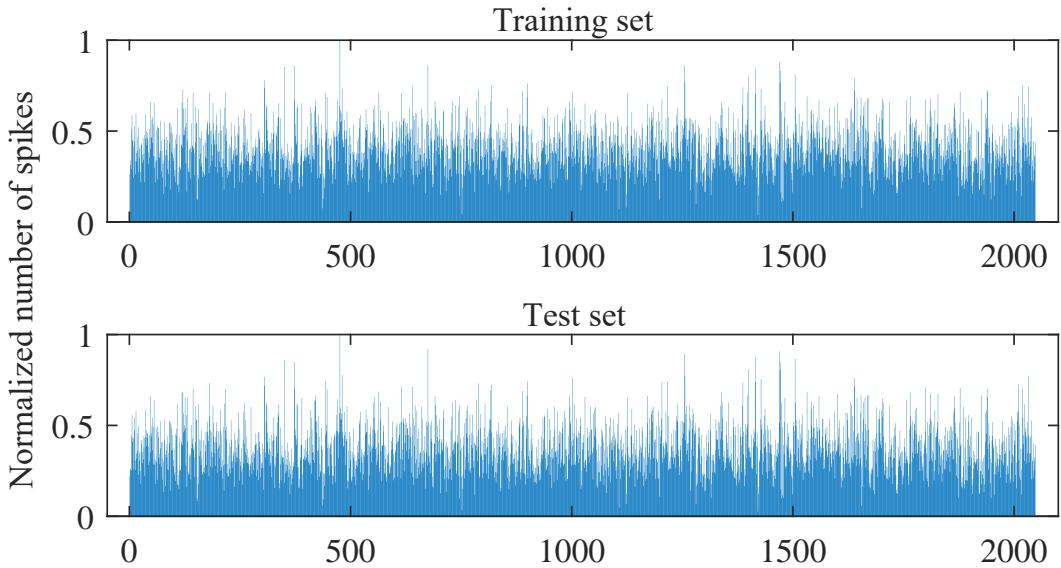


图 4.2 VGG11 网络的第二层全连接层在训练集和测试集上归一化后的输入分布。

Figure 4.2 The normalized input distribution of the second fully-connected layer of VGG11 in training set and test set.

的分布在一层内是不同的，并且训练集和测试集之间的输入分布差异很小。

基于此，在这里本文认为，对于针对具有明确的对象的神经网络，在进行权重映射的时候需要同时对输入和权重进行考虑，并通过现有的训练数据集来估计可能的测试输入分布。这种方法并不仅仅适用于基于阻变存储器阵列的加速器芯片和脉冲神经网络，对于人工神经网络同样可以通过该方法进行调整。

4.1.3 针对权重排布算法的优化

除了针对输入的映射方法之外，本文还针对权重-交叉阵列的映射方法进行了改进。改进的动机来自于前人的工作，在他们的文章中提出的权重映射方法具有两个问题，1) 只做针对行或列进行排布；2) 排布的调整范围仅局限于整个权重，也即将整个权重视为一个交叉阵列来进行调整。本文称这种方法为“*In-array*”的排布方法。然而，这些方法忽略了一个事实，即由于可靠性的问题，交叉阵列不能做到无限大 **AI_CAS64**，且权值总是由交叉阵列 (Shafiee 等, 2016) 表示。因此，在交叉阵列内部的交换实际上是可操作的，如果仅考虑整体的排布，那么将为这个问题带来不必要的约束条件，基于此，本文提出了一种“*Cross-array*”的排布方法，以在阵列内部进行进一步调整，其对比如图 4.3 所示：

该图显示了“*In-array*”方法和我们的方法之间的区别。假设权重矩阵 W_0 可

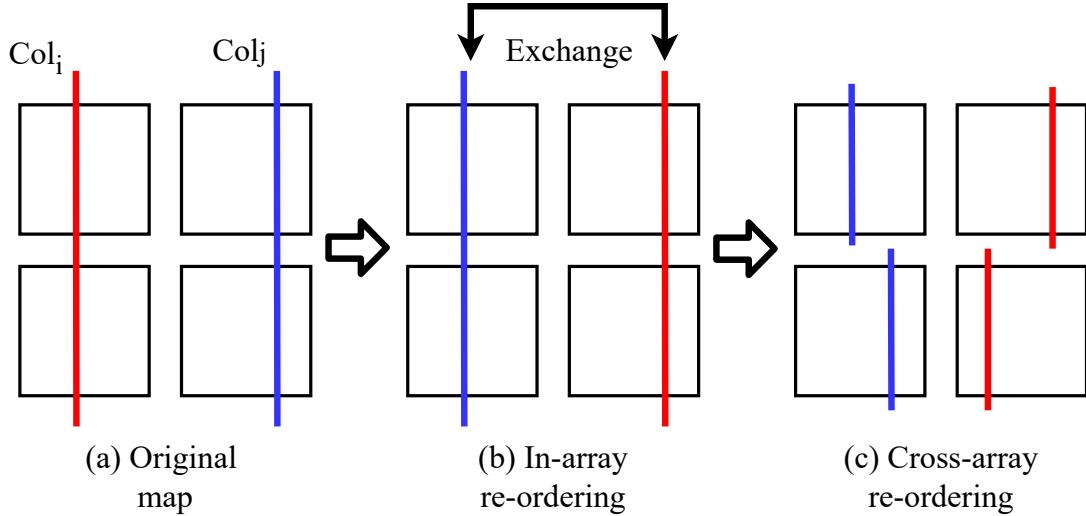


图 4.3 对权重矩阵进行不同的列排布方法的差异。

Figure 4.3 The column re-ordering difference.

以用 4 个交叉阵列表示，即图 4.3(a) 中所示的 4 个正方形。 Col_i (红线) 和 Col_j (蓝线) 表示权重 W_0 的第 i^{th} 和 j^{th} 列。然后可以看出，图 4.3(b) 中的 “In-array” 方法只能交换整个列的序列，即使上下两个阵列之间的列并无关系，下面的阵列的对应列也必须随上面的移动而移动，而图 4.3(c) 中的 “cross-array” 方法则具有与每个交叉阵列内的其他列进行交换的能力。因此，“Cross-array” 方法比 “In-array” 方法具有更多的解空间来获得更好的排布结果。

4.1.4 方法流程

本文提出了一种新的基于输入分布的排布和布局优化算法来均衡加速器芯片中交叉阵列中的功耗。本文首先提出了一种新的芯片布局方法，以避免将高功耗密度的器件过度集中，从而使得芯片的功耗分布更加均衡。此外，本文研究了在加速器芯片中功耗（热）问题中输入分布的影响，并利用这种分布来降低功耗对芯片的影响。最后，本文设计了一种有效的排布方法，该方法改进了前人的权重排布方法，通过考虑不同交叉阵列之间的行列交换，使得权重的排布工作更加精细。

本文的优化框架如图 4.4 所示，首先从基于输入和网络结构的阵列排布算法以及针对布局的优化算法对功耗进行均衡分布，其次，再做好分布后，使用 Hostspot 仿真软件及 (Beigi 等, 2018b) 提出的 endurance 曲线对优化后的方案进

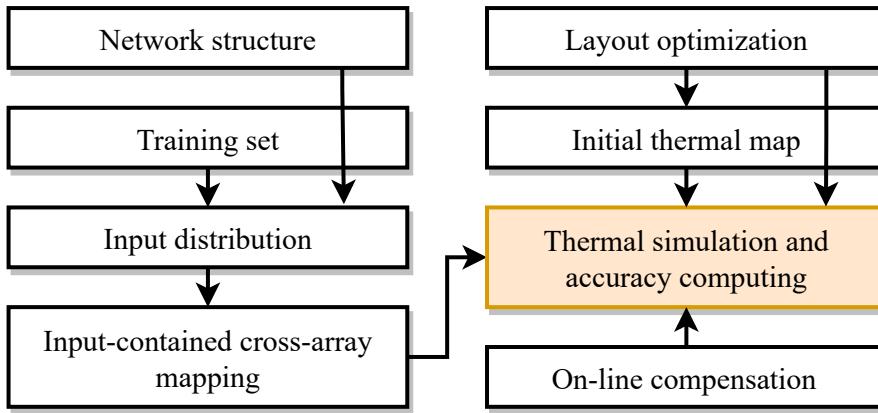


图 4.4 对交叉阵列功耗（热）问题的优化框架。

Figure 4.4 Proposed optimization framework.

行评估，最后采用 (Shin 等, 2020) 提出的在线补偿方法对该工作进行补偿以进一步减少功耗问题对芯片的影响。

4.2 结构调整

如前文所述，芯片上元件的布局将会影响到芯片内的功耗分布情况。因此，本文提出了一种启发式的布局优化方法以减少芯片结构对功耗分布的影响，其核心思想是通过将高功耗密度的原件分离开来达成均衡功耗分布的作用。由于该布局最终优化的目标是减少集中的功耗对芯片的影响，因此本文优先考虑功耗的分布问题，而对于其他评价指标，如面积、延迟等，本文假定芯片设计者能接受这些方面上增加的少量成本。

原始的 ISAAC 布局如图 4.5(a) 所示，其将多个交叉阵列 (Memristor Crossbar, XB) 以及多个数模转换器 (Digital-to-analog converter, DAC) 集中放置，由于 XB 和 DAC 是该芯片上功耗密度最大的两个元件 (Shafiee 等, 2016)，这两者的集中放置势必会导致功耗的集中。因此，本文提出的芯片布局如图 4.5(b) 所示，通过调整每个 IMA (In-situ Multiply Accumulate) 内部组件的布局来均衡每一个 IMA 内的功耗分布。在改进后的布局中，一个 TILE 包含 8 个 IMA 和相应的外围电路 (如输入输出寄存器，模数转换器等)，一个 IMA 包含 4 个相同的组成部分 (Constituent part of IMA, (CPI))。从图中可以看出，本文提出的布局方法将交叉阵列 (XB) 和数模转换器 (DAC) 分离，并将 2 个 DAC、2 个 ADC 和 2 个 XB 作为一个整体，以避免高功率密度组件的聚集。本文称这种布局为 “MA”。

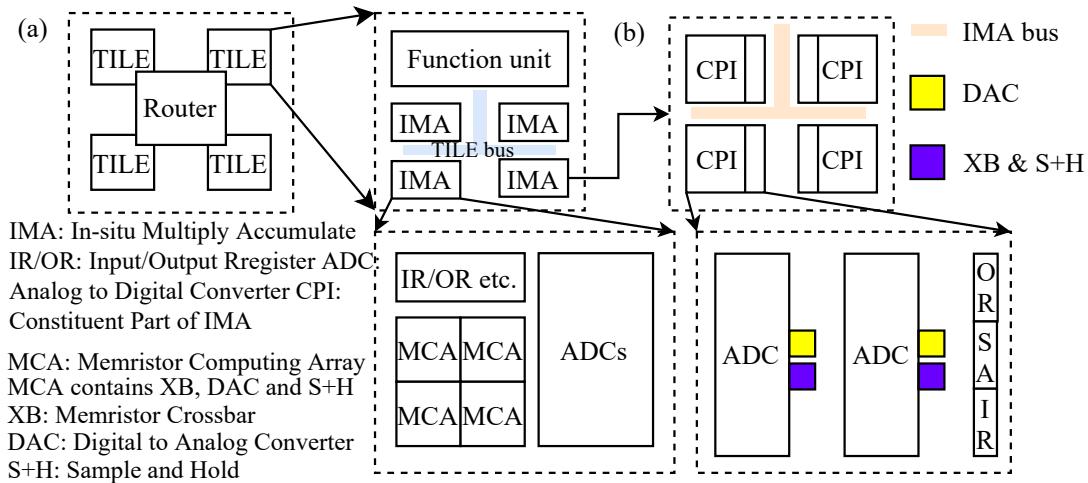


图 4.5 (a)ISAAC 结构。 (b) 改进后的 IMA 结构。

Figure 4.5 (a) ISAAC structure (Shafiee 等, 2016). (b) An example of our proposed layout-called “MA” for each IMA.

该布局方法不仅可以有效的将加速器芯片中高功率的部分进行分离，除此之外，这种布局还可以减少模拟数据的传输损伤。由于在 ISAAC 结构中，大部分数据传输的格式为相对模拟信号更加稳定的数字信号，而模拟信号的数据传输流仅存在于数模转换器 → 交叉阵列 → 模数转换器（Analog-to-digital Converter, ADC）中，所以，减少该部分的传输路径也会减少传输过程中对模拟信号造成的损伤。因此，由于本布局方法将图 4.5(a) 中相应的 DAC、XB 和 ADC 相结合，其相对传统的 ISAAC 结构来说可以缩短模拟信号的传输距离（特别是对于 XB 到 ADC 的传输路径），以实现减少传输损失的作用。

4.3 功耗分布

除了优化布局的方法外，均衡功耗分布的另一种方法是改变交叉阵列中的权重分布 (Beigi 等, 2018a; Shin 等, 2020)。基于 4.1.2 小节以及 4.1.3 小节的观察，本文提出了一种新的权重重排布工作流程，通过考虑输入分布和阻变存储器阵列之间的交换来均衡功耗分布，并降低热效应。该工作流程分为 3 个部分：首先，4.3.1 小节将介绍初始热图构建，其次，4.3.2 小节将阐述对输入的分布分析，最后，4.3.3 小节将介绍本文提出的的权重排布方法。

4.3.1 初始热图构建

如前文所述，芯片中的温度将对交叉阵列的状态具有很大影响，且阻变存储器的不同状态对温度的敏感性也不同。如图 2.10 所示，当温度升高时，LRS 变化较大，而 HRS 基本不变。因此，为了合理的映射交叉阵列内的权重，需要对阻变存储器可能的映射位置进行温度的预估计，以避免将较大的权值映射到高温区域。假设每个 TILE 内的初始环境温度和布局分别表示为 $T_{ambient}$ 和 $Layout_0$ ，且每个 TILE 中除阻变存储器阵列外的其他元件的功率记为 P_0 。然后，即可利用 Hotspot (W. Huang 等, 2006) 模拟在当前状态下的稳态温度图 TM_0 ，以此得到每个交叉阵列可能的放置位置的初始温度条件 T_{init} 。

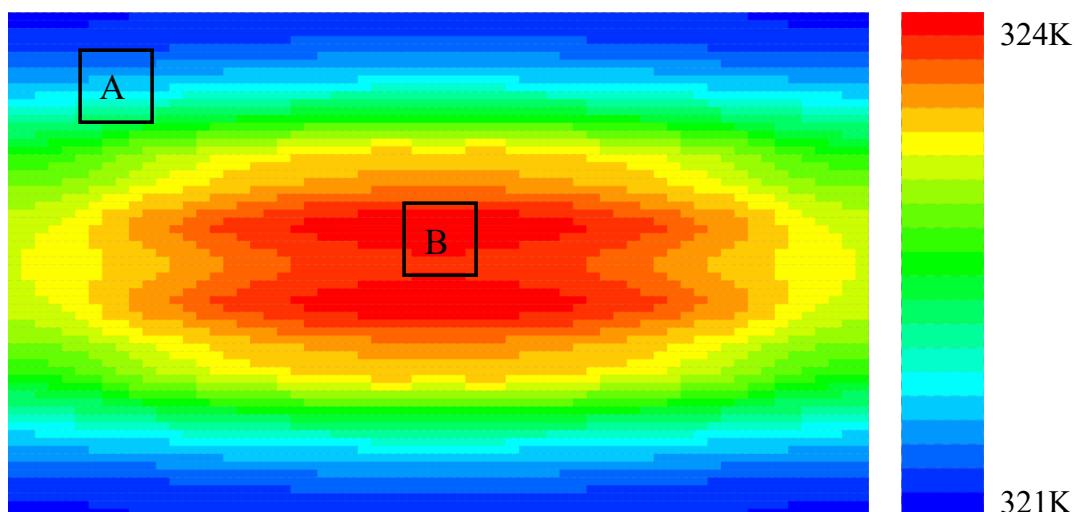


图 4.6 交叉阵列的初始热图。

Figure 4.6 An initial thermal map of memristor crossbar.

图 4.6 展示了一个示例 TILE 在进行 Hotspot 热仿真之后的结果，从图中可以看出，区域 A 的温度情况较区域 B 的温度情况更加良好，初始热更低。因此，在进行交叉阵列放置时，需要优先将可能产生较大功耗的交叉阵列放置到区域 A，而非区域 B。

4.3.2 对输入的分析

在 4.1.2 小节中，本文已展示了输入分布对功耗的影响，也即不同的输入状态会影响交叉阵列产生的功耗。因此，为了将该影响引入到对权重分布的建模中，本文根据阻变存储器的计算模式，首先根据训练数据对神经网络中每一层的

输入进行建模，得到在训练集下，每一层神经网络的输入 $Dist$ ；在此之后，对每一层的权重进行如下转换：

$$W_{bias_i} = Dist_i \otimes W_i \quad \dots (4.2)$$

$$\begin{bmatrix} x \\ y \end{bmatrix} \otimes \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} ax & bx \\ cy & dy \end{bmatrix} \quad \dots (4.3)$$

其中符号 W_{bias_i} 表示在神经网络第 i 层中考虑输入分布的权重矩阵， \otimes 表示一个新的乘法运算符，公式 (4.3) 具体展示了该运算符的操作方式。需要注意的是，尽管在公式 (4.1) 中，输入的平方和功耗是成正比关系，但由于在基于脉冲神经网络的芯片中，其网络的输入电压为恒定值，因此，在计算考虑输入的权重 W_{bias} 的时候，不需要对输入分布进行平方。

4.3.3 基于输入的权重排布方法

4.3.3.1 权重排布算法流程

在 4.3.1 小节和 4.3.2 小节中，本文对芯片的初始热效应和输入分布情况进行了分析，并获得了每一个 TILE 中，各交叉阵列的初始热效应图 T_{init} 和输入脉冲的分布情况 $Dist$ 。在这一小节中，本文提出了一种新的权重排布算法来进一步降低权重的分布对功耗的影响，其算法的实现过程如 Alg. 1 所示。

该算法的输入为初始热效应图 T_{init} 、输入脉冲分布 $Dist_i$ 、权重矩阵 W_i 和初始权重位置 L_i ，其中 $i \in \{1, 2, 3, \dots, n\}$ 表示对应元素在本算法所应用的 n 层的网络的第 i 层。

本文所提出的权重排布方法的过程可以分为 3 个部分。

第一部分是预处理部分，该算法利用根据训练集产生的输入脉冲分布来估计神经网路中每一层的真实输入分布；然后，该算法通过使用这个估计量以及公式 (4.2) 来获得包含输入的权重，如 Alg. 1 中第 3 行及公式 (4.2) 所示。

在预处理部分以及得到包含输入的权重 W_{bias} 之后，该算法的第二部分利用此权重和“Cross array”的排布方法进行权重的排布，以得到排布的权重和位置矩阵 W_{re} 和 L_{re} 。详细的排布算法见 4.3.3.2 小节和 Alg. 1 中的 RE-ORDER 函数。

第三部分是映射部分，该算法将包含输入的权重 W_{bias} 划分为 N 个大小为 $k \times k$ 的小权重矩阵 w ，并将它们映射到阵列中。其映射规则是，对于所有的 w_i , $i = 1, 2, \dots, N$, 值较大的 w_i 匹配 T_{init} 中温度较低的区域，反之亦然。其映射结果存储在 Map 中。

在运行完以上三个步骤后，本算法将得到最终的映射结果，并通过公式 (2.6) 和映射结果 Map 将权重矩阵编程到交叉阵列中。

4.3.3.2 权重排布算法

为了解释本文所提出的算法，本文首先从数学的角度对问题进行建模，其所需求解问题的数学模型如公式 (4.4) 所示：

$$\min (max(P) - min(P)) \quad \dots (4.4)$$

s.t. *Every row (or column) in w_i must be in the same row (or column) of origin weight;*

在这里, $P = \{p_1, p_2, \dots, p_N\}$, $p_i = w_i.sum()$, $i = 1, 2, \dots, N$ 。 w_i 表示每一个交叉阵列中映射的权重集合, p_i 表示每一个交叉阵列中权重集合的和, P 表示 p_i 的集合。公式 (4.4) 中的两个对行列的限制条件是为了保证原来在权重矩阵同一行 (列) 的权重，在调整后必须同样在一个 XB 内的同一行 (列)，该限制条件是为了确保 ISAAC 结构能够正常运行 (Shin 等, 2020)。

显而易见，该问题是一个 NP-Complete 问题，因此，本文在此使用了一种启发式的算法去近似求解该问题。具体的算法在 Alg. 1 中的 RE-ORDER 部分和图 4.7 中展示

图 4.7 展示了本文所提出的排布算法的工作流程。假定待排布的权重矩阵 W_{bias} 可以被分为 4 部分，如图 4.7-Step 1 所示，则本算法将首先在整体的权重矩阵层面进行行交换，通过不断迭代来降低这四个部分之间的预计功耗差异。该交换分为两步，第一步通过 BLDM 算法 (Shin 等, 2020) 进行粗略的交换，第二步在第一步的基础上，通过不断交换预计功耗最大的部分和最小部分的合适行，来达到不断缩小这四个部分之间预计功耗差的作用。第二步的迭代将在达到设

Algorithm 1 Re-ordering Process

Input: Initial thermal map T_{init} ; Input spikes $Dist$; Weight matrix W ; Initial weight location L ; Array size D ;

Output: Re-ordering weight matrix W_{re} ; Re-ordering location matrix L_{re} ; Weight-to-memristor map Map

```

1: function PROCESS( $T_{init}, Dist, W, L$ );
2:     % Pre-processing, get the input-considered weight
3:      $W_{bias\_i} \leftarrow Dist_i \otimes W_i$  for all  $i$ ;
4:     % Re-ordering
5:      $W_{re}, L_{re} = \text{RE-ORDER}(W_{bias}, L, Param)$ 
6:     % Mapping process
7:      $Map \leftarrow$  Based on  $L_{re}$  and  $T_{init}$ , map weights to the MCA by order;
8:     return  $W_{re}, L_{re}, Map$ 
9: end function
10: function RE-ORDER( $W_{bias}, L, Param$ );
11:      $Sum \leftarrow$  sum of rows or cols in  $W$ , the selection of row or col depends on
12:          $Param.type$ ;
13:      $Sort \leftarrow$  sort the  $Sum$  with descending order;
14:      $Part \leftarrow$  Divide the  $W_{bias}$  into several parts, and map each row or col to these
15:         parts by the order of  $Sort$ , as shown in Fig. 4.8.
16:     for  $i : 1 \rightarrow Param.iter$  do
17:         Find  $Part_{max}$  and  $Part_{min}$ , which have the largest (smallest) sum value of
18:              $Part$ .
19:          $diff \leftarrow Part_{max} - Part_{min}$ ;
20:         Find 1 row or col in  $Part_{max}, Part_{min}$ , respectively, and their difference is
21:             close to  $diff/2$ , exchange them.
22:     end for
23:     Repeat line 11 – 18 for each part and finer re-ordering as shown in Fig. 4.7.
24: end function

```

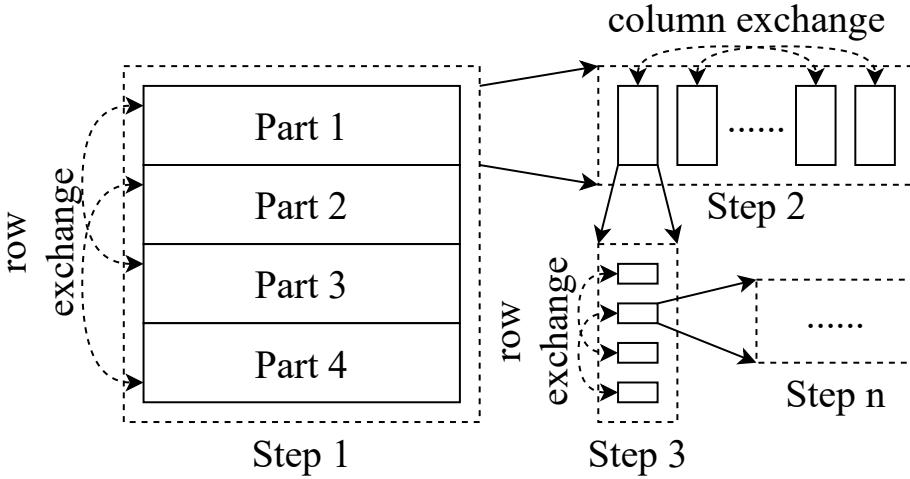


图 4.7 改进的权重排布算法。

Figure 4.7 Illustration of proposed re-ordering method.

定次数或者各个部分的功耗差低于一定阈值时停止，这两步的算法在 Alg. 1 中的第 11 – 13 行（第一步）和第 14 – 18 行（第二步）表现。在完成整体的交换后，该算法的下一步将会对每一个部分中的列进行交换，如图 4.7-Step 2 所示，交换方式同前文所述。以此类推，在设定的迭代次数后停止。

由于需要满足 ISAAC 结构计算的要求，这种交替交换的方式需要针对具体的交叉阵列的大小进行设计。本文使用的交叉阵列大小为 128×128 ，算法的总行列交换次数设定为 2 次（一次行一次列，列交换时最小单位设为 128）。以此来避免不满足公式 (4.4) 中限制条件的情况。

图 4.8 展示了本文所使用的 BLDM 算法的一个例子。假定一个权重矩阵 W_{bias} 具有 8 列，那么该算法将首先计算每一列的权重和，并存储在 col_sum 中。其次，定义初始位置向量 $location$ 以存储当前各列的位置。假定这里的优化目标时将该权重矩阵划分为两部分，那么该算法的第一步为选择权重和最大的前两列，并将其分别放入至不同的部分中，如图 4.8 中将权重和为 8 的列放在 $block_1$ 中，权重和为 7 的列放在 $block_2$ 中。此后，将在剩余的列中权重和最大的前两项选出，以和上一次相反的顺序放入至 $block_1$ 和 $block_2$ 中，如图 4.8 中将权重和为 5 的列放在 $block_1$ 中，权重和为 6 的列放在 $block_2$ 中，以此类推，直至分配完全部的列。在分配结束后，该算法得到最终分配到 $block_1$ 和 $block_2$ 中的列的结果，也即权重和为 1, 4, 5, 8 的列分配到 $block_1$ ，权重和为 2, 3, 6, 7 的列分配到 $block_2$ 。将 $block_1$ 和 $block_2$ 结合即得到该算法的输出—分配后的权重矩

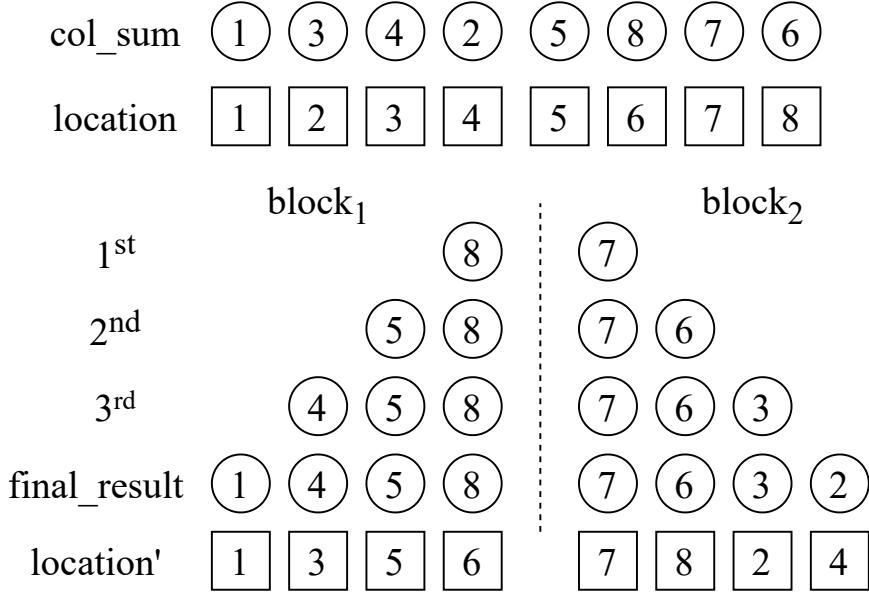


图 4.8 BLDM 算法示例

Figure 4.8 An example of BLDM algorithm.

阵 W_{re} 。此外，该算法还将得到该分配结果的对应位置矩阵 $location'$ ，记为 L_{re} 。在重映射时，可以根据 L_{re} 实现对权重的映射或者复原。

4.4 实验结果

4.4.1 实验设置

本文通过实验验证的方式评估了本文提出的热敏可靠性问题优化框架。本文所使用的基于阻变存储器的加速器芯片采用了 128×128 的阵列尺寸 (Shafiee 等, 2016)。此外，为了满足 ISAAC 的工作条件 (Shafiee 等, 2016)，这里本文将每个阻变存储器的分辨率设置为 2 位。在此设置下，本文所模拟的加速器芯片可以满足 ISAAC 的工作流程。所以，本文将芯片的工作频率设置为 $1.2GHz$ (Shafiee 等, 2016)，阻变存储器的电阻范围为 $[5K\Omega, 500K\Omega]$ ，输入电压范围为 $[0V, 0.9V]$ ，其余参数的功耗面积指标如表 4.1 所示。在对功耗的分析以外，为了更好的评估功耗对加速器芯片造成的影响，本文使用芯片的热仿真器 Hotspot (W. Huang 等, 2006) 来模拟功耗对片上温度的影响。此外，本文假设基于阻变存储器的加速器完全运行，并且卷积层的计算是并行的。在该模拟实验中，本文还假设不同 TILE 之间的效果可以被忽略。热仿真器 Hotspot 的其他详细参数，包括散热片参数等，设置方法与此前的工作 (Beigi 等, 2018a, b; Zhou 等, 2019) 相同，环

表 4.1 ISAAC 中部分参数。

Table 4.1 Some parameters of ISAAC.

Component	Param	Spec	Power	Area(mm^2)
ADC	resolution	8 bits	16mW	0.0096
	number	8		
DAC	resolution	1 bit	4mW	0.00017
	number	8×128		
S+H	number	8×128	$10\mu W$	0.00004
S+A	number	4	0.2mW	0.00024
IR	size	128 B	$77.5\mu W$	0.00013
OR	size	256 B	0.23 mW	0.00005

境温度设置为 300K (Shin 等, 2020)。为了保证对比的合理性, 其他被本文复现的方法所采用的单个 TILE 面积与本文提出的方法的单个 TILE 的面积是相同的 ($0.4055mm^2$)。

本文使用 PyTorch 框架 (Paszke 等, 2019) 分别在 CIFAR10 (对于 *VGG9* 网络) 和 CIFAR100 (对于 *VGG11* 网络) 数据集 (Krizhevsky 等, 2009) 上对从 *VGG9* 和 *VGG11* 转换而来的两个脉冲神经网络模型进行了验证。这些 SNN 模型通过 ANN 到 SNN 的转换 (Diehl 等, 2015) 和脉冲反向传播算法 (Spike Timing Dependent Backpropagation, 简称 STDB) 方法得到, 该方法在 (Rathi 等, 2020) 中实现。对比的设计配置如下:

- *Base*: 将神经网络权重映射到交叉阵列的直接映射。
- *MA*: 本文提出的对交叉阵列的布局修改方案。
- *TOPAR-C*: 前人 (Shin 等, 2020) 提出的基于平衡最大一阶差分方法 (Balanced largest first differencing method, 简称 BLDM) 的列排布方法。
- *MP-FLP*: 本文提出的所有方法的综合 (包括芯片布局和重新排序方法)。

4.4.2 实验结果及分析

图 4.9 表示在 *VGG11*、*VGG9* 网络中, 不同方法使用过后的归一化功率范围 (图 4.9(a)) 和神经网络中不同类型层的平均功率范围的减小量 (图 4.9(b))。将

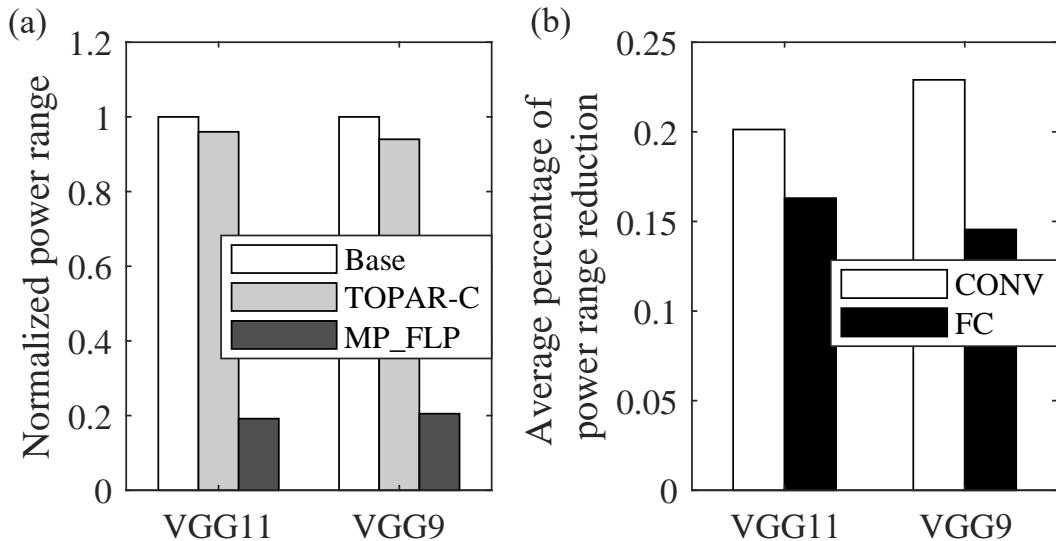


图 4.9 (a) 标准化的功率范围, (b) 卷积和全连接层的功率范围减少的幅度。

Figure 4.9 (a) The normalized power range. (b) The average percentage of power range reduction.

排布后的权值映射到每个交叉阵列后, 本文通过加载测试集来计算每个阵列的平均功率, 并采用不同的方法记录其功率范围, 其计算方法如公式 (4.5) 所示:

$$\text{功率范围} = \text{当前 } TILE \text{ 上, 交叉阵列的最大功率} - \text{最小功率} \quad \dots (4.5)$$

图 4.9(a) 中这些条形图的高度表示这些范围的标准化值 (以未调整的功率范围作为标称值)。从该图中可以发现, 与 TOPAR-C 相比, 本文所提出的方法在 *VGG11* 和 *VGG9* 上都获得了更好的性能。图 4.9(b) 比较了本文的方法在卷积 (CONV) 层和全连接 (FC) 层之间的性能差异。从图中可以发现, 该方法在卷积层中平均缩短了约 20%, 且在全连接层中平均缩短了约 15%。据此本文得出, 本文提出的方法在两种情况下均能取得较好效果, 虽然在不同类型的网络结构中方法的效果有差异, 但相距不大。

图 4.10 展示了 *VGG9* 网络中第三层全连接层最热的 TILE 的温度分布。从图中可以发现, (1) 通过使用 *MA* 方法, 芯片上的温度分布相对 *Base* 情况更加均匀, 且最高温度和使用 *TOPAR-C* 方法的芯片已十分接近; (2) 通过结合 *MA* 方法和排布方法, 最终使用的 *MP-FLP* 方法相比 *Base* 情况和前人的工作 (*TOPAR-C*) 更能有效降低峰值温度。

不同方法在 *VGG11* 和 *VGG9* 网络的峰值温度情况如图 4.11(a) 所示。从图中可以发现, 本文所提出的方法可以比 *TOPAR-C* 降低更多的峰值温度。在

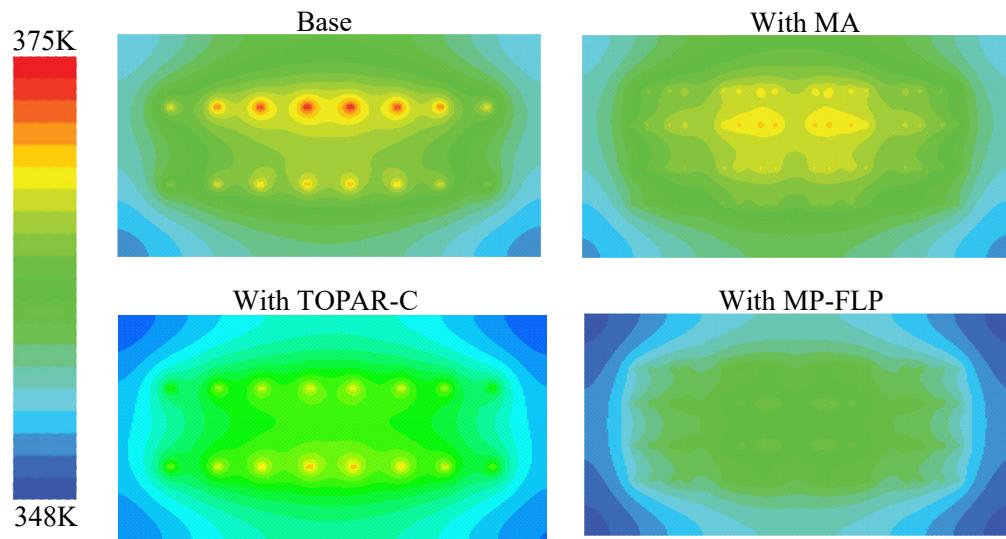


图 4.10 映射到加速器芯片的 $VGG9$ 网络中最热的 TILE 的温度分布

Figure 4.10 The temperature distribution of the hottest TILE in the third fully-connected layer of $VGG9$.

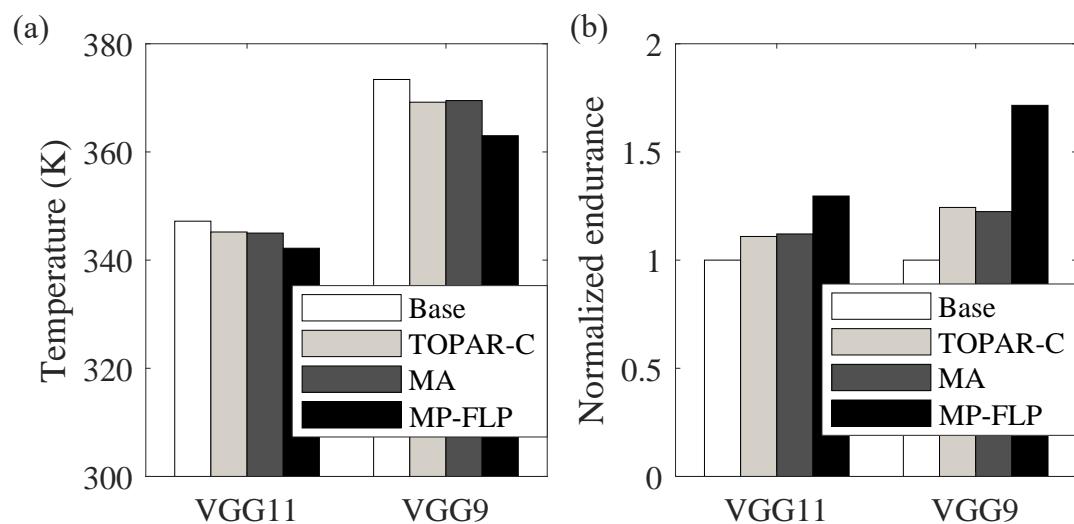


图 4.11 $VGG11$ 和 $VGG9$ 在不同方法下的 (a) 最高温度比对图。 (b) 标准化后的使用寿命比对图。

Figure 4.11 (a) Peak temperature and (b) Normalized endurance, of $VGG11$ and $VGG9$ in different methods.

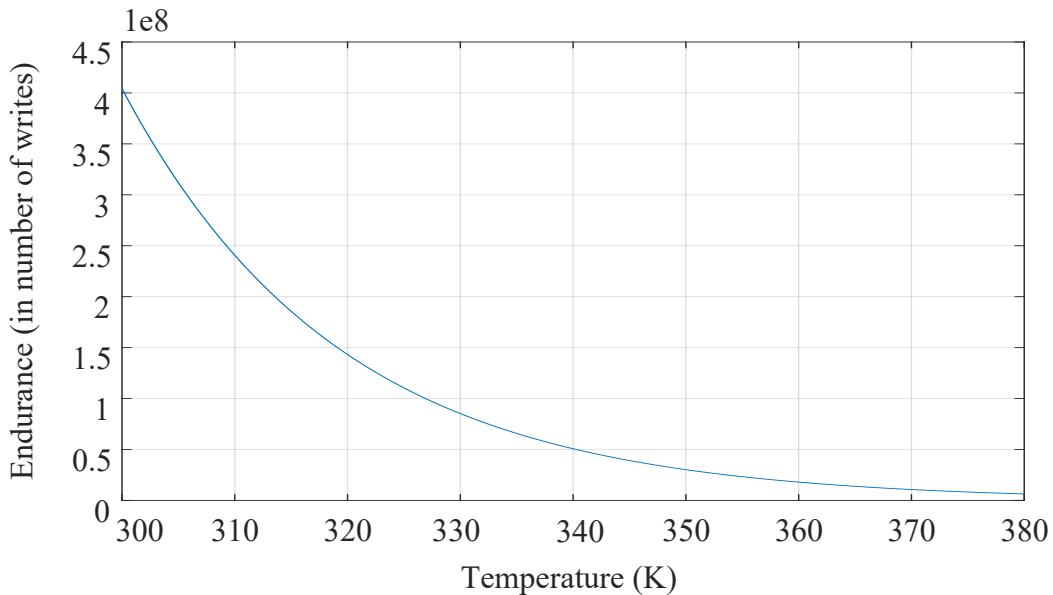


图 4.12 阻变存储器的使用寿命随温度的变化曲线 (Beigi 等, 2018b)。

Figure 4.12 Endurance-temperature curve of memristor (Beigi 等, 2018b).

VGG11 和 *VGG9* 网络中的最高峰值温度降低分别为 $5.0K$ 和 $10.4K$ 。由于 *TOPARC* 方法在 *VGG11* 和 *VGG9* 的最大峰值温度降低分别为 $2.1K$ 和 $4.2K$, 因此, 本文提出的方法较前人的方法在均衡功耗分布, 降低温度方面效果更好。

峰值温度的降低延长了阻变存储器的使用寿命 (Endurance) (Beigi 等, 2018b; Shin 等, 2020)。对于使用寿命的评估, 本文根据 (Beigi 等, 2018b), 假定阻变存储器在环境温度为 $300K$ 的情况下, 初始可被写次数为 4.14×10^8 , 以不同温度下阻变存储器可被写入的次数来验证使用寿命的增强。阻变存储器的使用寿命与温度的展示图如图 4.12 所示, 其横坐标表示不同温度情况, 纵坐标表示在当前温度下, 阻变存储器可被写入的次数。由于在 (Shin 等, 2020) 中, 阻变存储器的寿命被定义为第一次出现故障的时间, 因此本文将用芯片上各个交叉阵列的峰值温度来评估使用寿命的延长 (Shin 等, 2020)。其使用寿命比较如图 4.11(b) 所示。与 *Base* 情况相比, 本文提出的方法可以在 *VGG11* 和 *VGG9* 中分别提高 $1.30\times$ 和 $1.72\times$ 的使用寿命。

4.5 本章小结

针对基于阻变存储器的加速器芯片的热敏可靠性问题, 本文提出了三种对芯片上交叉阵列进行功耗分布优化的算法, 其分别是对芯片的结构调整、对神

经网络输入分布的考虑以及对权重排布算法的优化。通过这三种方法，本文均衡了交叉阵列上的功耗。实验结果表明，本文提出的算法在 VGG11 和 VGG9 中均取得了很好的效果，分别在这两个网络上降低了 5.0K 以及 10.4K 的峰值温度降低，而前人提出的算法仅能分别降低 2.1K 以及 4.2K。此外，通过对阻变存储器使用寿命的分析，本文提出的算法可以使得阻变存储器的使用寿命在这两个网络上分别延长至 1.3 倍和 1.72 倍。

第 5 章 总结与展望

5.1 总结

基于阻变存储器的加速器芯片具有较好的人工智能应用前景，然而其存在的可靠性问题限制了该加速器的发展。因此，本文针对加速器芯片上的供电网络可靠性问题以及加速器中阻变存储器交叉阵列的可靠性问题进行了研究。

在第一章中，本文首先对当前芯片工艺的技术瓶颈进行简单介绍，然后阐述当前芯片发展的三个主要趋势，由此引入基于阻变存储器的加速器芯片并介绍目前在该加速器的设计及使用过程中存在的可靠性问题。

在第二章中，本文首先对神经网络的基本概念和阻变存储器进行介绍，然后介绍阻变存储器在神经网络加速器中的应用以及其可靠性问题，最后对加速器上的供电网络分析进行介绍。

在第三章中，本文针对加速器芯片上的供电网络可靠性问题进行分析。基于对目前供电网络分析趋势的观察，本文找到了前人提出的层级结构分析方法的缺点，并加以改进。此后，本文根据该加速器芯片多同构核的实际情况，提出了针对多同构核的供电网络分析加速算法。实验结果表明，本文提出的方法可以在 8 核同构的加速器芯片上达到 7.17 倍的分析加速以及 99.9% 的精度，且该加速倍数与同构核数成正比关系。而在近同构的 8 核加速器芯片上，本文的方法可达到最高 5.21 倍的分析加速以及 99.9% 的精度。

在第四章中，本文针对加速器芯片中交叉阵列上的热敏可靠性问题进行改进。根据对前人提出的权重排布方法的分析总结，本文首先从结构调整的方向考虑，通过调整加速器芯片的结构，使得芯片结构对功耗的影响降低，此后，本文通过考虑神经网络的输入分布以及对交叉阵列进行权重排布的限制，提出改进的权重排布方法以均衡交叉阵列上的功耗分布。实验结果证明了本文提出方法的有效性，本文在 VGG9 和 VGG11 两个脉冲神经网络上分别比对了本文提出的方法和前人提出的方法。实验结果表明，本文所提出的方法可在 VGG11 和 VGG9 网络上分别降低 5.0K 和 10.4K 的加速器芯片峰值温度，以及分别延长 1.3 倍和 1.72 倍的加速器使用寿命。

5.2 展望

本文提出的方法虽然可以加速基于阻变式存储器的神经网络加速器的供电网络分析以及优化片上交叉阵列的功耗分布，但是依然存在一些局限性。本节在此对这些局限性进行总结，作为未来的研究工作。

1. 本文提出的供电网络分析方法，目前仅适用于静态的供电网络分析，暂未对动态工作条件的加速进行分析和实验。在后续的工作中，本文将尝试把静态方法推广至动态进行分析。此外，尽管本文的实验已说明本文方法的加速效果与供电网络的量级关系较小，但由于本文考虑的供电网络的量级较低，因此在后续工作中，本文将尝试对更大规模的供电网络进行分析。
2. 本文提出的结构调整方法，目前仅考虑了对功耗的均衡效果，未对其他方面（如延迟，面积等）进行评估，在后续的工作中，本文将尝试对这些指标进行分析，并尝试在考虑这些指标的情况下提出更全面的结构调整方案。
3. 在针对阻变存储器的功耗分布中，本文仅考虑了本文提出的方法与前人方法的比较，并未比较两种方法结合后的效果。由于其他优化功耗分布的方法同样具有很好的效果，因此本文在后续工作中将尝试将这些方法结合，以进一步解决阻变存储器的功耗分布问题。

参考文献

- HUAWEI, 2020. 半导体制程工艺发展史 [Z]. <https://forum.huawei.com/enterprise/zh/thread-705899.html>.
- 陈煌, 祝永新, 田犁, 等, 2018. 基于 FPGA 的卷积神经网络卷积层并行加速结构设计 [J]. 微电子学与计算机, 35(10): 85-88.
- 刘婷婷, 2011. 电源分配网络分析及电容器精确建模 [D]. 西安电子科技大学.
- ABIODUN O I, JANTAN A, OMOLARA A E, et al., 2018. State-of-the-art in artificial neural network applications: A survey[J]. *Helion*, 4(11): e00938.
- AGARAP A F, 2018. Deep learning using rectified linear units (relu)[J]. ArXiv preprint arXiv:1803.08375.
- AMBROSI E, BRICALLI A, LAUDATO M, et al., 2019. Impact of oxide and electrode materials on the switching characteristics of oxide ReRAM devices[J]. *Faraday discussions*, 213: 87-98.
- AMIRSOLEIMANI A, ALIBART F, YON V, et al., 2020. In-Memory Vector-Matrix Multiplication in Monolithic Complementary Metal–Oxide–Semiconductor-Memristor Integrated Circuits: Design Choices, Challenges, and Perspectives[J]. *Advanced Intelligent Systems*, 2(11): 2000115.
- AYERS J E, 2018. Digital integrated circuits: analysis and design[M]. CRC Press.
- BEIGI M V, MEMIK G, 2018a. Thermal-aware optimizations of ReRAM-based neuromorphic computing systems[C]//Proceedings of the 55th Annual Design Automation Conference: 1-6.
- BEIGI M V, MEMIK G, 2018b. THOR: THermal-aware Optimizations for extending ReRAM lifetime[C]//2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS): 670-679.
- BUSHNELL M, AGRAWAL V, 2004. Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits[M]. Springer Science & Business Media.
- CHEN A, 2016. A review of emerging non-volatile memory (NVM) technologies and applications[J]. *Solid-State Electronics*, 125: 25-38.
- CHI P, LI S, XU C, et al., 2016. Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory[J]. *ACM SIGARCH Computer Architecture News*, 44(3): 27-39.
- COWAN J D, 1990. Discussion: McCulloch-Pitts and related neural nets from 1943 to 1989[J]. *Bulletin of mathematical biology*, 52(1): 73-97.

- DENNARD R H, GAENSSLEN F H, YU H N, et al., 1974. Design of ion-implanted MOSFET's with very small physical dimensions[J]. IEEE Journal of solid-state circuits, 9(5): 256-268.
- DIEHL P U, NEIL D, BINAS J, et al., 2015. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing[C]// 2015 International joint conference on neural networks (IJCNN): 1-8.
- ESMAEILZADEH H, BLEM E, AMANT R S, et al., 2011. Dark silicon and the end of multicore scaling[C]// 2011 38th Annual international symposium on computer architecture (ISCA): 365-376.
- FAN C N, ZHANG F Y, 2011. Homomorphic filtering based illumination normalization method for face recognition[J]. Pattern Recognition Letters, 32(10): 1468-1479.
- FANG Y C, LIN H Y, SU M Y, et al., 2018. Machine-learning-based dynamic IR drop prediction for ECO[C]// Proceedings of the International Conference on Computer-Aided Design: 1-7.
- FANG Z, WANG X, LI X, et al., 2013. Fully CMOS-compatible 1T1R integration of vertical nanopillar GAA transistor and oxide-based RRAM cell for high-density nonvolatile memory application[J]. IEEE transactions on electron devices, 60(3): 1108-1113.
- GEPNER P, KOWALIK M F, 2006. Multi-core processors: New way to achieve high system performance[C]// International Symposium on Parallel Computing in Electrical Engineering (PAR-ELEC'06): 9-13.
- GHOSH-DASTIDAR S, ADELI H, 2009. Spiking neural networks[J]. International journal of neural systems, 19(04): 295-308.
- GOULDING N, SAMPSON J, VENKATESH G, et al., 2010. GreenDroid: A mobile application processor for a future of dark silicon[C]// 2010 IEEE Hot Chips 22 Symposium (HCS): 1-39.
- HENNESSY J L, PATTERSON D A, 2019. A new golden age for computer architecture[J]. Communications of the ACM, 62(2): 48-60.
- HINTON G E, OSINDERO S, TEH Y W, 2006. A fast learning algorithm for deep belief nets[J]. Neural computation, 18(7): 1527-1554.
- HO C W, RUEHLI A, BRENNAN P, 1975. The modified nodal approach to network analysis[J]. IEEE Transactions on circuits and systems, 22(6): 504-509.
- HU M, STRACHAN J P, LI Z, et al., 2016. Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication[C]// 2016 53nd acm/edac/ieee design automation conference (dac): 1-6.

- HUANG J J, TSENG Y M, LUO W C, et al., 2011. One selector-one resistor (1S1R) crossbar array for high-density flexible memory applications[C]//2011 international electron devices meeting: 31-7.
- HUANG W, GHOSH S, VELUSAMY S, et al., 2006. HotSpot: A compact thermal modeling methodology for early-stage VLSI design[J]. IEEE Transactions on very large scale integration (VLSI) systems, 14(5): 501-513.
- JOARDAR B K, LI B, DOPPA J R, et al., 2019. REGENT: A heterogeneous ReRAM/GPU-based architecture enabled by NoC for training CNNs[C]//2019 Design, Automation & Test in Europe Conference & Exhibition (DATE): 522-527.
- KARNEZOS M, 2004. 3D packaging: Where all technologies come together[C]//IEEE/CPMT/SEMI 29th International Electronics Manufacturing Technology Symposium (IEEE Cat. No. 04CH37585): 64-67.
- KHAN H N, HOUNSHELL D A, FUCHS E R, 2018. Science and research policy at the end of Moore's law[J]. Nature Electronics, 1(1): 14-21.
- KIM Y B, 2010. Challenges for nanoscale MOSFETs and emerging nanoelectronics[J]. Transactions on Electrical and Electronic Materials, 11(3): 93-105.
- KOZHAYA J N, NASSIF S R, NAJM F N, 2002. A multigrid-like technique for power grid analysis[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 21(10): 1148-1160.
- Krizhevsky, et al., 2009. CIFAR-10 and CIFAR-100 datasets[J].
- LEE C, PANDA P, SRINIVASAN G, et al., 2018. Training deep spiking convolutional neural networks with stdp-based unsupervised pre-training followed by supervised fine-tuning[J]. Frontiers in neuroscience, 12: 435.
- LI T, BI X, JING N, et al., 2017. Sneak-path based test and diagnosis for 1R RRAM crossbar using voltage bias technique[C]//Proceedings of the 54th Annual Design Automation Conference 2017: 1-6.
- LIU X, ZHOU M, ROSING T S, et al., 2019. HR 3 AM: a heat resilient design for RRAM-based neuromorphic computing[C]//2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED): 1-6.
- LIU X, MAO M, LIU B, et al., 2016. Harmonica: A framework of heterogeneous computing systems with memristor-based neuromorphic computing accelerators[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 63(5): 617-628.

- MA X, 2013. An overview of recent developments and applications of the GMRES method[J]. Pure Math, 3: 181-187.
- MA Y, ZHOU P, 2021a. Efficient Techniques for Training the Memristor-based Spiking Neural Networks Targeting Better Speed, Energy and Lifetime[C] // 2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC): 390-395.
- MA Y, ZHOU P, 2021b. Efficient Techniques for Training the Memristor-based Spiking Neural Networks Targeting Better Speed, Energy and Lifetime[C] // 2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC): 390-395.
- MOORE G E, 1998. Cramming more components onto integrated circuits[J]. Proceedings of the IEEE, 86(1): 82-85.
- MOURGIAS-ALEXANDRIS G, TSAKYRIDIS A, PASSALIS N, et al., 2019. An all-optical neuron with sigmoid activation function[J]. Optics express, 27(7): 9620-9630.
- NASSIF S R, 2008. Power grid analysis benchmarks[C] // 2008 Asia and South Pacific Design Automation Conference: 376-381. DOI: [10.1109/ASPDAC.2008.4483978](https://doi.org/10.1109/ASPDAC.2008.4483978).
- NICKOLLS J, DALLY W J, 2010. The GPU computing era[J]. IEEE micro, 30(2): 56-69.
- NITHIN S K, SHANMUGAM G, CHANDRASEKAR S, 2010. Dynamic voltage (IR) drop analysis and design closure: Issues and challenges[C] // 2010 11th International Symposium on Quality Electronic Design (ISQED): 611-617. DOI: [10.1109/ISQED.2010.5450515](https://doi.org/10.1109/ISQED.2010.5450515).
- PASZKE A, GROSS S, MASSA F, et al., 2019. Pytorch: An imperative style, high-performance deep learning library[J]. Advances in neural information processing systems, 32.
- QIAN H, SAPATNEKAR S, 2004. Hierarchical random-walk algorithms for power grid analysis[C] // ASP-DAC 2004: Asia and South Pacific Design Automation Conference 2004 (IEEE Cat. No.04EX753): 499-504. DOI: [10.1109/ASPDAC.2004.1337626](https://doi.org/10.1109/ASPDAC.2004.1337626).
- QUARTERLY T, 2016. AFTER MOORE'S LAW[Z]. <https://www.economist.com/technology-quarterly/2016-03-12/after-moores-law>.
- RATHI N, SRINIVASAN G, PANDA P, et al., 2020. Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation[J]. ArXiv preprint arXiv:2005.01807.
- SAAD Y, 1993. A flexible inner-outer preconditioned GMRES algorithm[J]. SIAM Journal on Scientific Computing, 14(2): 461-469.
- SHAFIEE A, NAG A, MURALIMOHAR N, et al., 2016. ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars[J]. ACM SIGARCH Computer Architecture News, 44(3): 14-26.

- SHIN H, KANG M, KIM L S, 2020. A thermal-aware optimization framework for reram-based deep neural network acceleration[C]//Proceedings of the 39th International Conference on Computer-Aided Design: 1-9.
- SUTTER H, et al., 2005. The free lunch is over: A fundamental turn toward concurrency in software[J]. Dr. Dobb's journal, 30(3): 202-210.
- TAVANAEI A, GHODRATI M, KHERADPISHEH S R, et al., 2019. Deep learning in spiking neural networks[J]. Neural networks, 111: 47-63.
- TAYLOR M B, 2012. Is dark silicon useful? Harnessing the four horsemen of the coming dark silicon apocalypse[C]//DAC Design Automation Conference 2012: 1131-1136.
- VATAJELU E I, DI NATALE G, ANGHEL L, 2019. Special session: Reliability of hardware-implemented spiking neural networks (SNN)[C]//2019 IEEE 37th VLSI Test Symposium (VTS): 1-8.
- WALCZYK C, WALCZYK D, SCHROEDER T, et al., 2011. Impact of temperature on the resistive switching behavior of embedded HfO_2 -Based RRAM devices[J]. IEEE transactions on electron devices, 58(9): 3124-3131.
- Wikipedia, n.d. Neuron[Z]. <https://en.wikipedia.org/wiki/Neuron>.
- WONG H S P, LEE H Y, YU S, et al., 2012. Metal–oxide RRAM[J]. Proceedings of the IEEE, 100(6): 1951-1970.
- WULF W A, MCKEE S A, 1995. Hitting the memory wall: Implications of the obvious[J]. ACM SIGARCH computer architecture news, 23(1): 20-24.
- XIE Z, REN H, KHAILANY B, et al., 2020. PowerNet: Transferable dynamic IR drop estimation via maximum convolutional neural network[C]//2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC): 13-18.
- XU B, WANG N, CHEN T, et al., 2015. Empirical evaluation of rectified activations in convolutional network[J]. ArXiv preprint arXiv:1505.00853.
- XUE C J, ZHANG Y, CHEN Y, et al., 2011. Emerging non-volatile memories: Opportunities and challenges[C]//Proceedings of the seventh IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis: 325-334.
- YAO P, WU H, GAO B, et al., 2017. Face classification using electronic synapses[J]. Nature communications, 8(1): 1-8.
- YE W, LI M, ZHONG K, et al., 2018. Power Grid Reduction by Sparse Convex Optimization[C]//Proceedings of the 2018 International Symposium on Physical Design: 60-67.

- YU S, 2018. Neuro-inspired computing with emerging nonvolatile memories[J]. Proceedings of the IEEE, 106(2): 260-285.
- YU S, GAO B, FANG Z, et al., 2012. A neuromorphic visual system using RRAM synaptic devices with sub-pJ energy and tolerance to variability: Experimental characterization and large-scale modeling[C]// 2012 International Electron Devices Meeting: 10-4.
- ZANGENEH M, JOSHI A, 2012. Performance and energy models for memristor-based 1T1R RRAM cell[C]// Proceedings of the great lakes symposium on VLSI: 9-14.
- ZHANG B, UYSAL N, FAN D, et al., 2019. Handling stuck-at-fault defects using matrix transformation for robust inference of dnns[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 39(10): 2448-2460.
- ZHANG G L, LI B, ZHU Y, et al., 2021. Robustness of Neuromorphic Computing with RRAM-based Crossbars and Optical Neural Networks[C]// Proceedings of the 26th Asia and South Pacific Design Automation Conference: 853-858.
- ZHANG S, ZHANG G L, LI B, et al., 2019. Aging-aware lifetime enhancement for memristor-based neuromorphic computing[C]// 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE): 1751-1756.
- ZHAO M, PANDA R V, SAPATNEKAR S S, et al., 2002. Hierarchical analysis of power distribution networks[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 21(2): 159-168.
- ZHIRNOV V V, CAVIN R K, MENZEL S, et al., 2010. Memory devices: Energy–space–time trade-offs[J]. Proceedings of the IEEE, 98(12): 2185-2200.
- ZHOU M, IMANI M, GUPTA S, et al., 2019. Thermal-Aware Design and Management for Search-Based In-Memory Acceleration[C]// Proceedings of the 56th Annual Design Automation Conference 2019: 1-6.

作者简历及攻读学位期间发表的学术论文与研究成果

已发表（或正式接受）的学术论文：

1. Zhang Chengrui, Ma Yu and Zhou Pingqiang, "Thermal-Aware Layout Optimization and Mapping Methods for Resistive Neuromorphic Engines," 2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC), 2022, pp. 50-55.
2. Zhang Chengrui, Zhou Pingqiang. Improved Hierarchical IR Drop Analysis in Homogeneous Circuits[C]//2020 IEEE 15th International Conference on Solid-State & Integrated Circuit Technology (ICSICT). IEEE, 2020: 1-3.
3. Ma Yu, Zhang Chengrui and Zhou Pingqiang. Efficient Techniques for Extending Service Time for Memristor-based Neural Networks[C]//2021 IEEE Asia Pacific Conference on Circuit and Systems (APCCAS). IEEE, 2021: 81-84.

