

# 基于多值阻变存储器与子句折叠策略的并行梯度布尔可满足性求解器

岳文硕<sup>1,2</sup>, 叶林韬<sup>3</sup>, 史代璟<sup>1,2</sup>, 符一涵<sup>1,2</sup>, 杨宇翔<sup>1</sup>, 赵洪霄<sup>1,2</sup>, 李嘉怡<sup>1,2</sup>, 张 腾<sup>1</sup>,  
燕博南<sup>1,2</sup>, 杨玉超<sup>1,2,4,5\*</sup>

(1. 北京大学集成电路学院/集成电路高精尖创新中心新基石科学实验室, 北京 100871; 2. 北京大学人工智能研究院, 北京 100871;  
3. 北京大学信息科学技术学院, 北京 100871; 4. 北京大学信息工程学院广东省存算一体芯片重点实验室新基石科学实验室, 广东  
深圳 518055; 5. 北京脑科学与类脑研究所, 北京 102206)

**摘 要:** 本研究提出了一种基于存内计算技术和多值阻变存储器的并行梯度硬件布尔可满足性求解器——Fold-SAT。如其名称所示, Fold-SAT将多个子句折叠存储到存内计算阵列的单个列中, 即子句折叠策略。存内计算可以利用其高计算并行度对可满足性问题求解进行硬件加速。传统的存内可满足性求解器每列仅存储一个子句, 其存储阵列稀疏度较高, 常常占用较大存储空间。在迭代过程中, 存内计算加速器通过阵列上的乘累加计算实现对子句可满足性的计算。本工作通过多电阻态表征不同子句的结构信息, 可以在阵列中使用一列存储多个子句的信息, 并在读出阶段解码得到各个子句对应的可满足性结果。并行梯度计算是一种面向可满足性求解的高效算法, 其前向传播阵列与反向传播阵列均可使用子句折叠策略进行压缩。为了探究子句折叠策略对于可满足性问题的压缩存储效果, 本研究面向并行梯度计算设计了阵列中多值存储的电阻比例并研究其对数据压缩的影响。结果表明, 当前向传播阵列采用1:4:16的电阻态比例时, 该方法能将存储阵列规模缩小至传统方案的1/3, 并将总体存储稀疏度从97.0%降低至约92.8%, 其中存储稀疏度定义为阵列中不存储有效信息的单元占比。此外, 本研究还开发了在反向传播阵列中的数据压缩方案, 通过多值映射可以将反向传播阵列所需阵列规模缩小至传统方案的1/2。本研究探讨了器件非理想效应及参数设计所带来的影响, 仿真结果表明, 器件编程与读出需要将器件偏差控制在1  $\mu$ S以内, 电导均值偏移需要控制在基准值的10%以内。而反向传播阵列中的多值器件具有一系列可选择范围, 多值器件范围在1:8~1:16之间可实现合理的计算精度。仿真结果显示, 本工作 $N=50$ 的问题平均求解时间约为19.74  $\mu$ s, 与当前领先的加速器速度接近, 说明本工作可以在保持较高的计算速度同时缩小其所需存储器阵列规模。

**关键词:** 可满足性; 阻变存储器; 存内计算; 数据压缩; 多值存储

**基金项目:** 国家重点研发计划 (No.2023YFB4502200); 国家自然科学基金 (No.92164302, No.92364102, No.T2350006); 广东省存算一体芯片重点实验室 (No.2024B1212020002); 北京市自然科学基金 (No.L234026, No.L257010); 高等学校学科创新引智计划(111计划)(No.B18001); 本研究得到了新基石科学基金会的资助

**中图分类号:** TP302.1; TP389.1 **文献标识码:** A **文章编号:** 0372-2112(XXXX)XX-0001-12

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.12263/DZXB.20250966

## Multilevel Resistive Memory Enabled Parallel-gradient Boolean Satisfiability Solver with Clause-folding Scheme

YUE Wenshuo<sup>1,2</sup>, YE Lintao<sup>3</sup>, SHI Daijing<sup>1,2</sup>, FU Yihan<sup>1,2</sup>, YANG Yuxiang<sup>1</sup>, ZHAO Hongxiao<sup>1,2</sup>, LI Jiayi<sup>1,2</sup>,  
ZHANG Teng<sup>1</sup>, YAN Bonan<sup>1,2</sup>, YANG Yuchao<sup>1,2,4,5\*</sup>

(1. *New Cornerstone Science Laboratory, Beijing Advanced Innovation Center for Integrated Circuits, School of Integrated Circuits, Peking University, Beijing 100871, China;*

*2. Institute for Artificial Intelligence, Peking University, Beijing 100871, China;*

*3. School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China;*

*4. New Cornerstone Science Laboratory, Guangdong Provincial Key Laboratory of In-Memory Computing Chips, School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, Shenzhen, Guangdong 518055, China;*

*5. Chinese Institute for Brain Research, Beijing 102206, China)*

**Abstract:** This work introduces a hardware parallel-gradient boolean satisfiability solver, Fold-SAT, enabled by compute-in-memory technology and multilevel resistive memory storage. As shown in its name, Fold-SAT folds multiple clause-

es into one column of the compute-in-memory array. Compute-in-memory can utilize its computing parallelism to accelerate the solving process of satisfiability problems on hardware. A conventional in-memory SAT-solver only stores one clause per column. It shows high sparsity when stored in memory and usually occupies large memory capacity. During computing iterations, the compute-in-memory accelerator conducts multiply-and-accumulate operations on the array to calculate the satisfiability result of each clause. Fold-SAT uses multiple resistance states to represent the structural information of different clauses. It stores the information of multiple clauses in each column and decodes the satisfiability results of the clauses during readout process. Parallel gradient algorithm is an efficient algorithm for satisfiability problems. The clause-folding scheme applies to both the forward-pass array and the backward-path array in parallel gradient algorithm. To explore the compression result of Fold-SAT, this work designs the resistance ratio of multilevel storage and examines its effects on data compression. Results reveal that, when the forward-pass array uses a resistance ratio of 1:4:16, it reduces the array size to 1/3 of conventional methods and reduces the overall storage sparsity from 97.0% to about 92.8%. The storage sparsity refers to the portion of cells that do not store effective information. Moreover, this work explores the data compression in the backward-pass array. It reduces the required array size to half that of conventional methods. This work discusses the impacts of device non-idealities. Simulated results show that the device-programming algorithms should reduce the device errors to less than 1  $\mu$ S, and the deviation of conductance mean values should be controlled within 10%. There is a series of design spaces regarding the multilevel devices for the backward-pass array. It could reach acceptable computing precision when the resistance ratio is set between 1:8 to 1:16 for the backward-pass array. Simulation results show that this work achieves a median solving time of 19.74  $\mu$ s, which is close to the leading work in the literature. This illustrates that this work could reduce the memory capacity cost while maintaining comparatively high solving speed.

**Keywords:** satisfiability; resistive memory; in-memory computing; data compression; multilevel storage

**Foundation Item(s):** National Key Research and Development Program of China (No.2023YFB4502200); National Natural Science Foundation of China (No.92164302, No.92364102, No.T2350006); Guangdong Provincial Key Laboratory of In-Memory Computing Chips (No.2024B1212020002); Beijing Natural Science Foundation (No.L234026, No.L257010); 111 Project (No.B18001); This work has been supported by the New Cornerstone Science Foundation

## 0 引言

布尔可满足性问题(Satisfiability, SAT)是一类非确定性多项式时间(Non-Deterministic Polynomial, NP)完全问题<sup>[1-2]</sup>。许多实际问题均可转化为SAT问题<sup>[3]</sup>,例如形式化验证<sup>[4-5]</sup>以及计算机辅助设计(Computer-Aided Design, CAD)中的多种问题<sup>[6-7]</sup>。布尔可满足性问题旨在判断是否存在一组布尔变量的赋值方案,能够满足给定的布尔公式。该公式以合取表达式(Conjunctive Normal Form, CNF)表示,由多个子句(clause)的合取(逻辑与)构成,每个子句则是文字析取(逻辑或)的形式——文字即变量(variable)或其否定形式。在包含 $N$ 个变量和 $M$ 个子句的 $k$ -SAT问题中,CNF公式包含 $M$ 个子句,每个子句恰好包含 $k$ 个文字,且所有不重复变量总数为 $N$ 。

过去几十年来,SAT求解器的发展取得了显著进展,其中基于软件的方法(如BerkMin等)<sup>[8-9]</sup>与基于硬件的方案(包括存内计算、循环神经网络、存内并行处理及阻变存储器阵列中的并行梯度计算)均获得重要突破<sup>[10-13]</sup>。然而,随着变量数或子句规模的增加,软件SAT求解器会呈现指数级的时间复杂度增长。尽管目前软件SAT求解器可以通过并行计算等方式降低时间开销,面向SAT领域的专用硬件加速器

凭借其高并行度和可重构性,展现出实现高速大规模SAT求解的潜力<sup>[14-16]</sup>。已有研究表明,硬件退火处理器能在保持高并行性与可扩展性的同时,有效加速NP问题的启发式算法。因此,开发基于专用集成电路的SAT求解器,在保证高可解性的同时实现面积效率、能耗与时序性能的优化,已成为当务之急。

部分硬件SAT求解器采用专用电路表征变量,另一些则通过存内计算阵列存储和处理子句。利用处理单元(Processing Element, PE)加速变量计算虽能有效提升运算速度,但需要将变量对应的所有子句存储于其专属处理单元中<sup>[17]</sup>。这种方案的面积成本较高,且单个处理单元可容纳的子句数量有限。基于存内计算阵列的方案采用列表示子句、行表征变量,通过高并行度加速乘加运算,并降低频繁数据搬移的开销<sup>[18-21]</sup>。该方法降低了单个变量的面积成本,可支持与存储器位宽相当的大规模子句数量。然而,由于 $k$ -SAT问题中每列仅需 $k$ 个有效存储单元,导致该方案存在存储稀疏性问题,造成大量面积浪费并限制集成密度。因此,亟需开发具备更高计算密度和更优可扩展性的硬件SAT求解器。

直观来看,采用多值(multilevel)存储单元存储子句可提升存内计算宏模块的存储与计算密度<sup>[22-25]</sup>。最新研究表明,基于多值阻变存储器的计算能进一步

提升存内计算模块的存储密度。然而与神经网络加速不同的是, SAT问题无法直接映射到连续整数层面。传统多值阻变存储器阵列的乘加运算结果仅能反映连续整数层面的计算, 关键点在于如何协调阻变存储器阵列的连续计算特性与 SAT 变量的离散化表征。

本研究提出创新性方法 Fold-SAT, 通过多值阻变存储器实现在单列中同时映射和处理多个独立子句。实验结果表明, 相较于传统映射策略, 该方案将单阵列所需器件数量降低至原来的 1/3 左右。通过软硬件协同优化, 本工作开发了具有自适应压缩功能的、可同时面向正向传播阵列与反向传播阵列的并行梯度 (parallel gradient) 计算压缩方案。

本研究在 40 nm 阻变存储器芯片上对提出的多级阻变存储器 SAT 求解器进行了系统级评估, 结果显示, 本方案实现了将存储稀疏度从 97.0% 降低至 92.8%, 其中存储稀疏度定义为阵列中不存储有效信息的单元占比。本工作探讨了器件非理想效应对计算结果的影响。仿真实验显示, 器件编程与读出需要将器件偏差控制在  $1 \mu\text{S}$  以内, 电导均值偏移需要控制在基准值的 10% 以内。所提出的范式通过单列多子句映射机制, 为提升硬件 SAT 求解器的可扩展性开辟了新路径。

## 1 Fold-SAT 设计

本工作以领域内较为先进的并行梯度计算方法为基础<sup>[13]</sup>, 开发出具备自适应压缩机制的 Fold-SAT 方法。其计算流程如图 1 所示, 在计算开始前, 需要先将子句写入存内计算阵列并初始化变量。求解过程由多个循环迭代周期组成, 每个循环迭代周期包括前向传播计算与反向传播计算。前向传播计算用于得到在给定变量下各个子句的 SAT 数量。这一步可以在存内计算阵列中并行计算, 计算结果以电流的形式经由模数转换器 (Analog-to-Digital Converter, ADC) 读出, 并经由数字电路进行解码, 得到数学意义上的 SAT 数量。在正向传播计算结束后, 系统判断各个子句是否都实现满足, 即各个子句的 SAT 数量是否大于 0。如果全部实现满足, 那么求解过程结束。SAT 数量定义为一个子句中满足的文字的数量。如果一个子句中的 SAT 数量大于 0, 那么该子句就处于满足的状态; 如果一个子句中所有文字都未实现满足, 则该子句的 SAT 数量为 0, 子句处于未满足的状态。

如果未实现全部子句满足, 则进入反向传播计算阶段。反向传播的目的是得到下一代周期所要翻转的变量。反向传播阶段以子句为输入, 经由存内计算阵列, 而输出端为各个文字。这一步是为了统计在

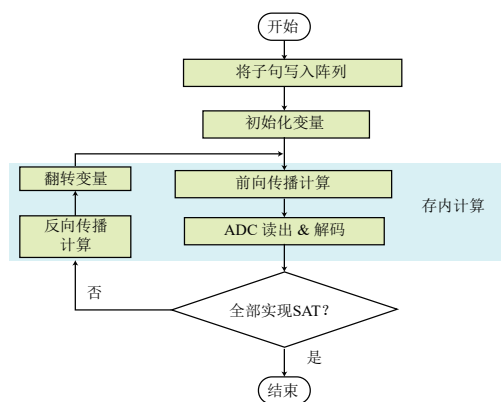


图1 SAT求解器工作流程图

Figure 1 Schematic of the SAT-solving workflow

当前条件下各个文字对应的“脆弱子句”数量。“脆弱子句”定义为 SAT 数量为 1 的子句。对于这样的子句, 只要其中实现满足的变量被翻转, 那么该子句的状态就会从满足变为未满足。反向传播阶段, 需要从当前未满足的子句中挑选变量来进行翻转。但同时需要考虑到各个变量所对应的“脆弱子句”数量, 应尽量挑选对应“脆弱子句”数量较少的变量来翻转, 以实现增加尽量多的满足子句。需要指出的是, 为了协助系统摆脱局部优解, 帮助系统更好地迭代, 翻转变量这一步常常也需要模拟退火和随机翻转算法来提供随机性。经过反向传播计算和翻转变量后, 系统挑选得到下一轮迭代所需要翻转的变量, 并进入下一轮迭代的前向传播计算。

Fold-SAT 有别于面向并行梯度计算的传统存内计算映射方法。Fold-SAT 主要将不同位线 (不同列) 上的权重通过映射到存储器不同的多值级别上的方式, 折叠到同一条位线上进行压缩存储。这一方案可以用于降低存储一个 SAT 问题所需的阵列器件数量, 并降低存储阵列的稀疏度。Fold-SAT 方案中, 同样包含前向传播阵列与反向传播阵列, 而本工作也研究了针对于前向传播阵列与反向传播阵列分别进行数据压缩的方法。

### 1.1 工作流程

总体而言, 存内 SAT 求解器使用存内计算操作来计算每一迭代周期 (iteration) 和每一问题 (case) 下的 SAT 数量。如图 2 所示, 一个基于阻变存储器的存内计算阵列可以存储一个给定 SAT 问题下的多个子句。阻变存储器阵列的一行代表一个变量  $V_i$ , 并且用它紧接着的下一行表示该变量的取反  $\bar{V}_i$ 。阻变存储器阵列的一列代表一个子句  $C_j$ 。如果变量  $V_i$  在子句  $C_j$  中, 那么阵列中在第  $i$  行与第  $j$  列的交叉点的器件就会被设置为低阻 (Low-Resistance State, LRS) 状态。如

果不是这样,那么相对应的器件则会保持高阻(High-Resistance State, HRS)状态。以 3-SAT 问题中的一个子句为例,如果其 CNF 形式的表达式为  $C_1 = V_1 \vee \overline{V_3} \vee V_7$ 。在这一例子中,在  $C_1$  列的 3 个器件都会被设置为低阻,其位置分别是  $(V_1, C_1)$ 、 $(\overline{V_3}, C_1)$  与  $(V_7, C_1)$ 。当使用存内计算 SAT 求解器来求解 SAT 问题时,阻变存储器阵列主要用于计算每个迭代周期下的 SAT 数量。首先,当前的每个变量及其取反会作为输入信号被送入阵列的每一行。然后,利用基尔霍夫定律,阵列每一列的输出电流为其上每个单元电流的加和。在阵列角度上,阻变存储器阵列会计算输入变量的向量与权重矩阵之间的乘累加(Multiply-and-Accumulate, MAC)操作。

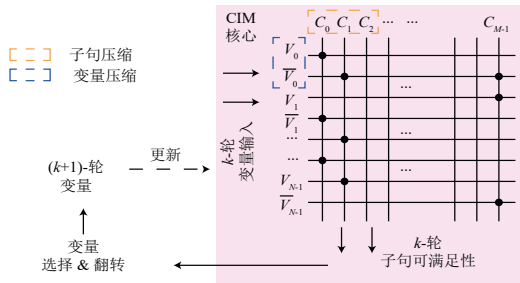


图2 前向传播阵列的工作机制

Figure 2 The schematic of the forward-pass array

最后,阵列中每一列的输出电流会被 ADC 采样并转化为数字信号。总体而言,每一个子句的 SAT 数量可以通过如下公式计算:  $SAT(C_j) = \sum_{i=0}^{N-1} V_i \wedge W(V_i, C_j) + \sum_{i=0}^{N-1} \overline{V_i} \wedge W(\overline{V_i}, C_j)$ 。在这一例子中,  $C_1$  列输出的 SAT 数量可以被计算为:  $SAT(C_1) = V_1 + \overline{V_3} + V_7$ 。当该 SAT 数量为 0 时,这一子句状态为不满足。当该 SAT 数量大于 0 时,这一子句状态为满足。

在这种传统映射方法下,以前向传播阵列为例,如果一个 SAT 问题具有  $N$  个变量与  $M$  个子句,那么该问题需要映射到一个  $2N \times M$  大小的存内计算阵列上,如图 2 所示。在其具有的  $M$  列中,每一列代表一个子句。而每一个变量及其取反各占据一行,共组成  $2N$  行。对于一个  $k$ -SAT 问题,每一列共有  $k$  个单元被置为有效,其余单元则在计算全程不被使用。

为了实现这样的计算模式,其前向传播部分需要使用如图 3 所示的硬件架构。该硬件架构包括一个阻变存储器存内计算阵列、位线(Bit-Line, BL)驱动电路、输入编码器、读出电路与模数转换器、SAT 解码器与数字逻辑等。而数字逻辑部分包括控制逻辑、随机

翻转逻辑与变量寄存器等。整体架构的适用性包含传统并行梯度计算与使用 Fold-SAT 压缩方法的并行梯度计算。其中,输入译码器与 SAT 解码器主要在 Fold-SAT 中使用。

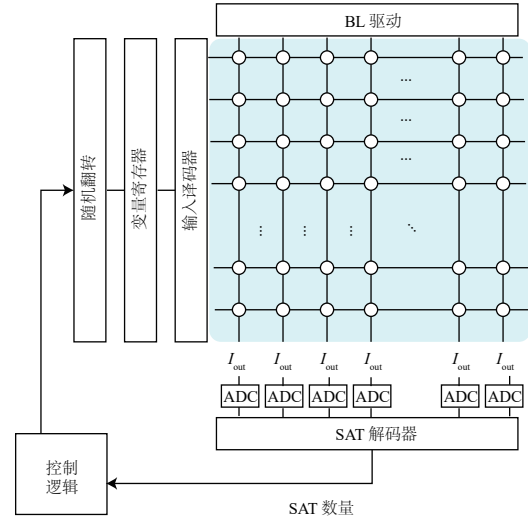


图3 Fold-SAT 中前向传播阵列的硬件示意图

Figure 3 The hardware schematic of the forward-pass array in Fold-SAT scheme

## 1.2 硬件映射

在本工作中, Fold-SAT 利用了阻变存储器的多值存储方法,将多个子句压缩存储至阵列里的一列中,即可以实现折叠策略。如图 4 所示, Fold-SAT 方法没有定义与 HRS 所对立的 LRS,而是针对 3 个不同的子句分别使用了 3 个阻态: RS0, RS1 与 RS2。在一个 3-SAT 问题中,可能的输出电流包括  $\{1 \times, 2 \times, 3 \times\} (V_{read}/R_{RS0})$ 、 $\{1 \times, 2 \times, 3 \times\} (V_{read}/R_{RS1})$  与  $\{1 \times, 2 \times, 3 \times\} (V_{read}/R_{RS2})$ 。只要不同状态的输出电流不相互重叠,这些状态就可以在阻变存储器的同一列里面表示不同的子句。通过这样,一列的输出电流可以表示为:  $I_j = \sum_{m=0}^2 V_{read}/R_{RSm}$   $\left[ \sum_{i=0}^{N-1} \overline{V_i} \wedge W(V_i, C_{3j+m}) + \sum_{i=0}^{N-1} V_i \wedge W(\overline{V_i}, C_{3j+m}) \right]$ 。通过特殊设计的模数转换器与 SAT 译码器, Fold-SAT 可以成功计算 3 个子句各自的 SAT 数量。

Fold-SAT 的硬件架构包含阻变存储器阵列、读出电路和数字逻辑模块,该架构降低了频繁数据传输的需求。变量存储于变量寄存器中,子句权重则通过多级存储方案本地保存在阻变存储器阵列内。在每次迭代中,输入解码器将变量送入阻变存储器阵列。阵列执行乘加运算后,将各列输出电流送至模数转换器。模数转换器将电流值转换为包含 3 个子句满足数之和的数字值。SAT 解码器从每列数字输出中分

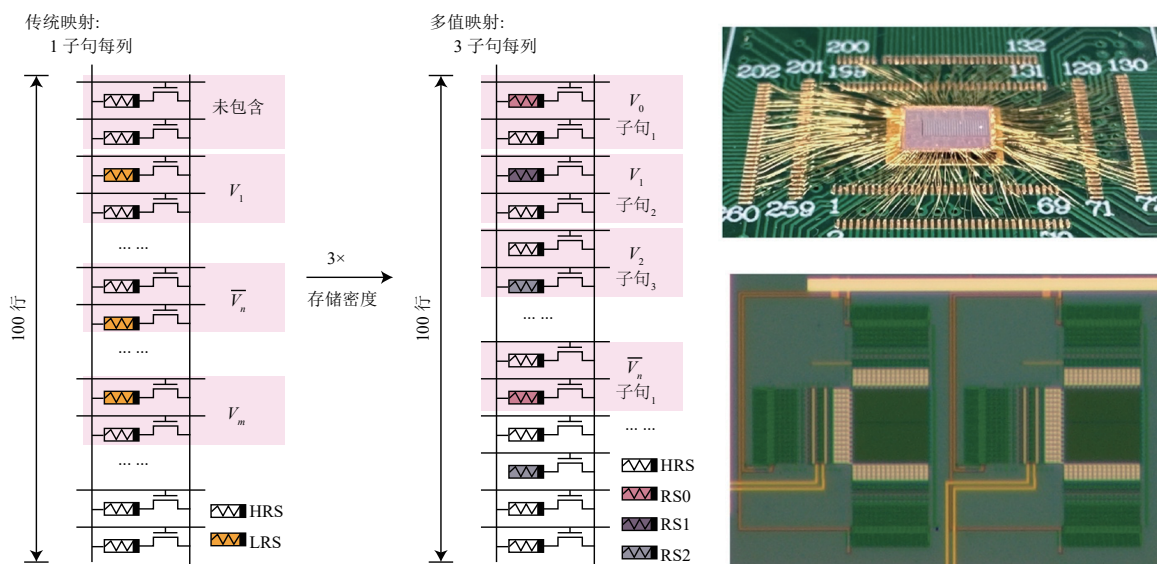


图4 3-SAT问题的多值映射示意图。右图为本工作使用的阻变存储器阵列

Figure 4 Schematic of the mapping of a 3-SAT problem on a multilevel RRAM array. Right part of the figure shows the RRAM array used in this work

离出3个子句的满足数值,至此完成单次迭代的满足数计算。随后控制逻辑执行并行梯度算法的控制流并与反向传播阵列相沟通,随机选择变量进行翻转操作。SAT求解过程通过迭代计算各子句满足数,并依据随机翻转算法管理变量翻转,直至问题满足可解条件,如图3所示。

通常而言,只要各输出端的电流分布互不重叠,阻变存储器阵列的单个列即可容纳多个子句。本研究通过实验与仿真验证了单列容纳3个子句的可行性。现以每列包含3个子句的Fold-SAT为例,阐述其电阻态的设计方法论。如图5所示,在本设计中,前向传播阵列的3种电阻状态的电阻比例被设置为  $R_{RS0}:R_{RS1}:R_{RS2}=16:4:1$ 。由此,当施加相同的读电压

$V_{read}$  在各个器件上时,其电流比值为  $I_{RS0}:I_{RS1}:I_{RS2}=1:4:16$ 。通过这一设计,可以使得3个子句的输出结果互不干扰。例如,Clause0结果的输出范围要小于单个Clause1器件所产生的电流。类似的,Clause1结果的输出范围小于单个Clause2器件所产生的电流。在数学上,如果定义数字化后的单列输出电流为  $Output$ ,那么 Clause0 的 SAT 数量可以用公式表示为:  $SAT(RS0) = Output \% 4$ ; Clause1 的 SAT 数量可以用公式表示为:  $SAT(RS1) = (Output \% 16) / 4$ ; Clause2 的 SAT 数量可以用公式表示为:  $SAT(RS2) = Output / 16$ 。这一方法可以降低存储 SAT 问题所需要的存储器开销,降低数据稀疏度,并减少存储 SAT 问题的存储冗余。

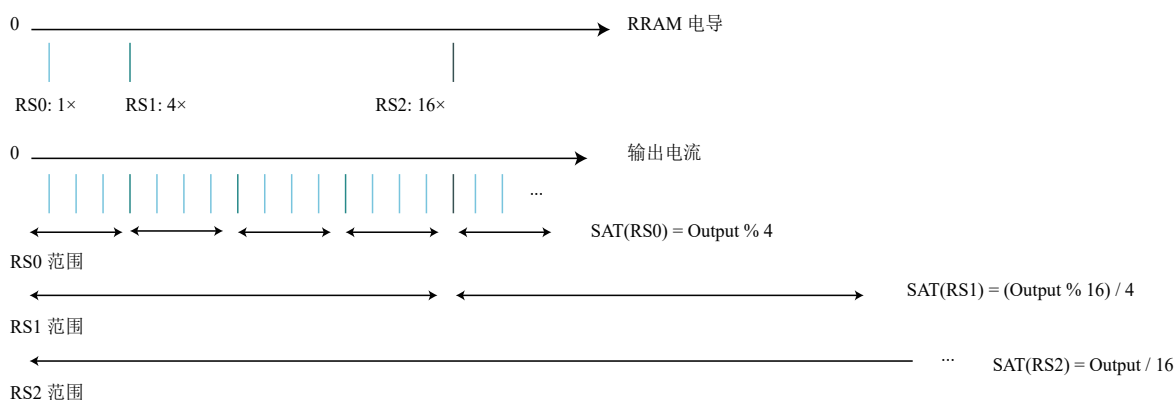


图5 压缩存储中各个电阻状态的大小关系

Figure 5 The magnitude relationships among the various resistance states in the compressed storage

如图6所示,使用Fold-SAT方法,在每列存储3个子句的情况下,存储前向传播阵列,可以降低存储器数量需求至原来的1/3左右。这一解码公式的适用范围为整数。在Fold-SAT应用过程中,需要使用模数转换器将阵列输出结果转化为整数,其对于阵列计算误差有一定要求,计算误差相关讨论在本文的结果与讨论部分进行展开。

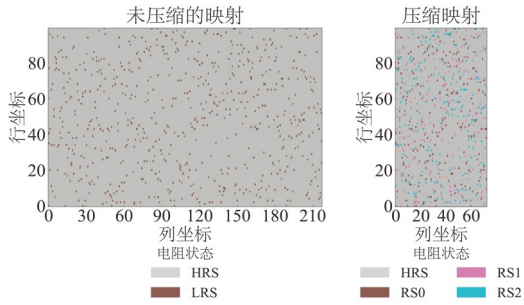


图6 压缩存储与传统存储方式的对比

Figure 6 The comparison between compressed storage and conventional storage scheme

### 1.3 Fold-SAT中的前向传播-反向传播协同压缩计算

在SAT求解的并行梯度方法中,前向传播阵列与反向传播阵列均可以使用Fold-SAT中的数据压缩方法进行压缩。如图7所示,并行梯度方法需要使用一个前向传播阵列与一个反向传播阵列协同进行计算。在传统映射下,如果一个问题包含 $N$ 个变量与 $M$ 个子句,那么前向传播阵列的大小为 $2N \times M$ ,而反向传播阵列的大小为 $M \times 2N$ 。两个阵列各自均需要存储全部子句权重信息。前向传播阶段,输入信息为各个文字的取值(由变量得到),输出结果为各个子句的SAT数量。反向传播阶段,输入信息为选中的子句,而输出结果为各个文字对应的“脆弱子句”数量。在同一迭代周期中,反向传播阶段对前向传播阶段具有数据依赖性。在已有工作中,前向传播计算和反向传播计算通过分别放置前向传播阵列和反向传播阵列两个阵列来实现,数据从前向传播阵列传输至反向传播阵列。

在本工作中,Fold-SAT方法针对前向传播阵列与反向传播阵列均可实现压缩存储,如图8所示,前向传播阵列的压缩方法如前所述,如果考虑3个子句经由多值存储方法存储在同一列,那么压缩后前向传播阵列的理论大小为 $2N \times M/3$ 。在反向传播阵列中,可以将变量 $V_i$ 及其取反 $\bar{V}_i$ 所对应的权重同时存储于同一列中。Fold-SAT方法同样使用多值存储方法来存储文字 $V_i$ 与 $\bar{V}_i$ 所对应的权重。设对应 $V_i$ 与 $\bar{V}_i$ 权重的电导比例为 $1:X$ ,那么 $X$ 有多种取值。需要指出的

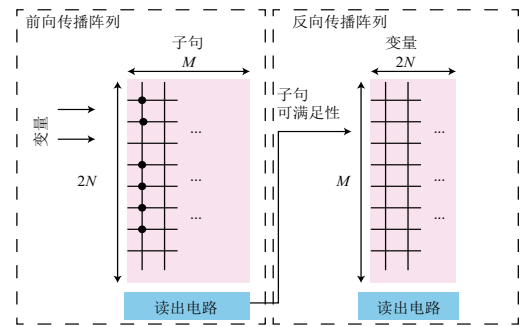


图7 完整运行算法所需的前向传播阵列与反向传播阵列

Figure 7 The forward-pass array and backward-pass array required by the complete algorithm

是,在一个SAT问题中,由于同一个文字可以出现在不定数量的子句中,在反向传播阵列的同一列里,一个文字对应权重出现的次数也是不定的。而在一次反向传播计算中,当 $V_i$ 积累的电流达到 $X$ 倍,也即超过单个有效 $\bar{V}_i$ 所产生的电流时,计算结果将出现错误。在本工作中,这一错误定义为变量错位。因此,若给定一个 $X$ 值,这一方法在反向传播中难以保证理论上的绝对精确。然而,存在一个 $X$ 值范围,能够使计算发生在容错范围内,在实践中可以达到较为合理的SAT求解结果。而不同的 $X$ 值,在读出电路方面需要使用不同精度的ADC来读出。当 $X$ 值越高,越不容易出现变量错位错误,而需要的读出电路精度也越高,所对应的功耗和面积开销也就越高;当 $X$ 值越低,需要的读出电路精度也越低,所对应的功耗和面积开销也越低,但更有可能出现变量错位错误。因此, $X$ 值的选择在实际情况下面临着取舍,存在一定设计空间。

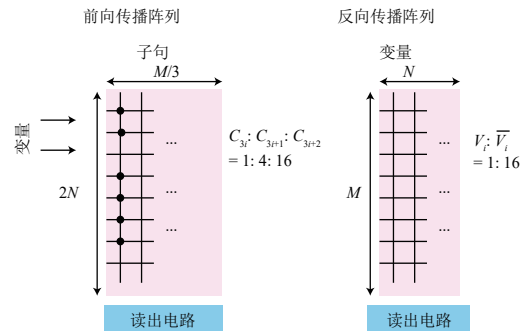


图8 前向传播阵列与反向传播阵列的压缩存储表示

Figure 8 The compressed storage of forward-pass array and backward-pass array

读出电路采用定制设计的逐次逼近型模数转换器与SAT解码器。如前所述,单个子句的乘加运算可产生4种输出电平(对应2 bit信息)。当多个子句折叠在一列时,乘加运算所产生的电平数也相应增加,

对应的模数转换器需要支持更多的读出电平数。

图9展示了读出电路基本结构:阻变存储器阵列完成模拟域乘加运算后,钳位电路将输出电流采样为电压值并传输至采样保持电路,随后模数转换器根据逐次逼近逻辑将模拟电压转换为数字量。在前向传播阵列的读出电路中,主要使用6 bit精度的模数转换器。而在反向传播阵列的读出电路中,可以在计算过程中适应性地选择模数转换精度。

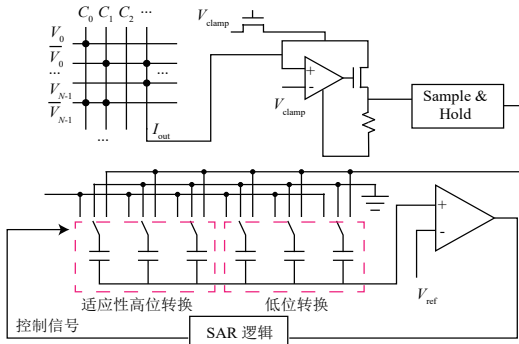


图9 读出电路示意图

Figure 9 The schematic of the readout circuit

## 2 结果与讨论

本研究在40 nm阻变存储器芯片上实验验证了Fold-SAT架构。该芯片包含多个典型尺寸为 $128 \times 128$ 的1T1R阵列。Fold-SAT设计方案对阻变存储器器件提出了特定要求:例如在实现每列包含3个子句的3-SAT问题时,前向阵列需要具备3种电阻态(RS0、RS1和RS2)。在数学上,对于包含 $N$ 个变量和 $M$ 个子句的问题,由于所需列数缩减至原来的 $1/3$ ,前向阵列尺寸为 $2N \times M/3$ 。反向阵列同样采用类似方法提升存储密度,其数学本质是前向阵列的转置矩阵。反向阵列中每行存储一个子句,Fold-SAT通过不同电阻值在单列中同时存储变量 $V_i$ 及其反变量 $\bar{V}_i$ 的子句权重,典型电阻比设置为 $1:16$ 。这种高电阻比设计能有效降低变量间串扰。相同问题规模下,反向阵列所需列数从 $2N$ 缩减至 $N$ (压缩为原来的 $1/2$ ),因此阵列尺寸为 $M \times N$ 。以50变量-218子句问题为例,在理论上,前向阵列尺寸为 $100 \times 73$ ,反向阵列尺寸为 $218 \times 50$ 。

为实现乘加运算中各电平无重叠,RS1电导值设为RS0的4倍,RS2电导值达到RS0的16倍。本研究最低阻态约为 $5 \text{ k}\Omega$ 。为直观展示多级阵列编程效果,我们采用基本的写入-验证算法在阵列层面实现了RS0、RS1和RS2的编程,如图10所示。电导分布直方图显示3个态在存在器件间波动时的分布情况:RS2、RS1和RS0的电导值分别为 $220.04 \mu\text{S}$ 、 $54.07 \mu\text{S}$ 和 $13.30 \mu\text{S}$ ,对应标准偏差为 $4.75 \mu\text{S}$ 、 $2.95 \mu\text{S}$ 和 $1.77 \mu\text{S}$ 。

制备的多级阻变存储器阵列在不同电导态下均表现出良好的保持特性,可满足Fold-SAT芯片的实际应用需求。在Fold-SAT实验实施过程中,观察到阵列层面的阻变存储器器件波动问题。本研究探索了多种优化方法,以提升计算精度并抑制器件间波动的影响。

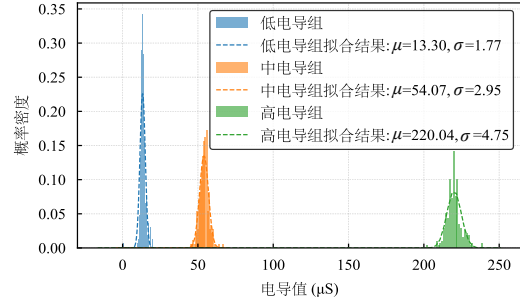


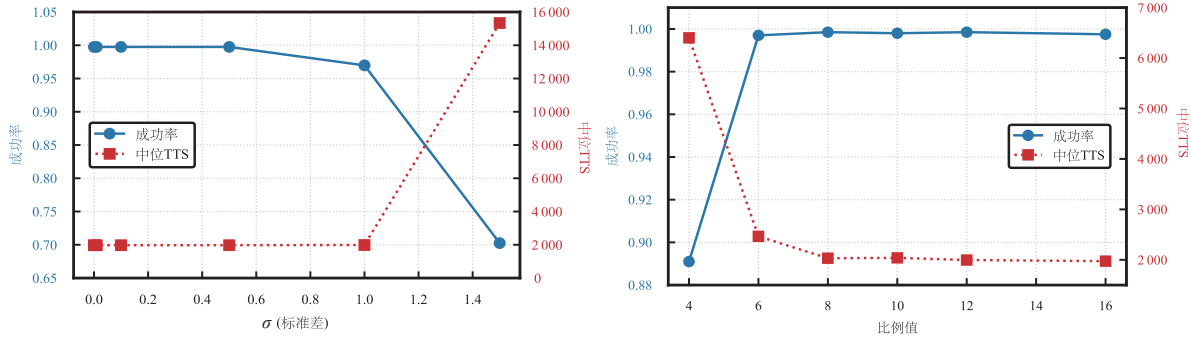
图10 在经历一般编程情况下的阵列电导分布

Figure 10 The conductance distribution of the RRAM array under normal programming conditions

在多次迭代计算过程中,阻变存储器的误差包括其编程误差以及多次读出误差。本工作测试了编程后阻变存储器的读出误差。读出误差与器件本身和测试系统均有关,除了器件本身的读出噪声外,读出误差中还可能引入来自测试板、电路走线、分立元件等带来的噪声。本工作针对于RS2、RS1、RS0 3种状态下的器件测试了读出噪声,每个器件进行 $2^{17}$ 次读出,实验测得其读出噪声标准差为 $2.33 \mu\text{S}$ 、 $1.37 \mu\text{S}$ 和 $0.62 \mu\text{S}$ 。而利用仿真采样综合两种噪声,其等效的整体标准差为 $5.39 \mu\text{S}$ 、 $3.31 \mu\text{S}$ 和 $1.94 \mu\text{S}$ 。读出噪声使得整体误差相对编程误差而言增加 $10\% \sim 13\%$ 。

性能仿真研究表明,Fold-SAT方案对阻变存储器电阻精度具有较高要求。仿真结果显示,当电导标准偏差超过 $1 \mu\text{S}$ 时,系统性能将出现退化。如图11(a)所示,在标准偏差小于 $1 \mu\text{S}$ 时,成功率和中位求解时间保持稳定;当标准偏差达到 $1.5 \mu\text{S}$ 时,成功率显著下降,中位求解时间显著增加。成功实现Fold-SAT需要将阻变存储器器件编程误差控制在 $1 \mu\text{S}$ 以内。由于阵列误差的来源包括编程误差以及读出噪声两方面,在计算中需要考虑降低这两种误差使得其低于要求所需误差。最新研究表明,高精度编程方法可实现 $2 \mu\text{S}$ 间隔的阻变存储器编程精度<sup>[26-27]</sup>,这一方案既包括更精确的写入-验证机制,也包括使用去噪声方法用于减少读取过程中的噪声干扰,从而提高阻变存储器的精度。这为Fold-SAT达到所需精度提供了技术可行性。

反向阵列的读出精度同样影响系统性能。读出精度与反向阵列的电阻比相关, $1:16$ 的电阻比可有效避免变量与其反变量权重间的串扰,但需要5 bit

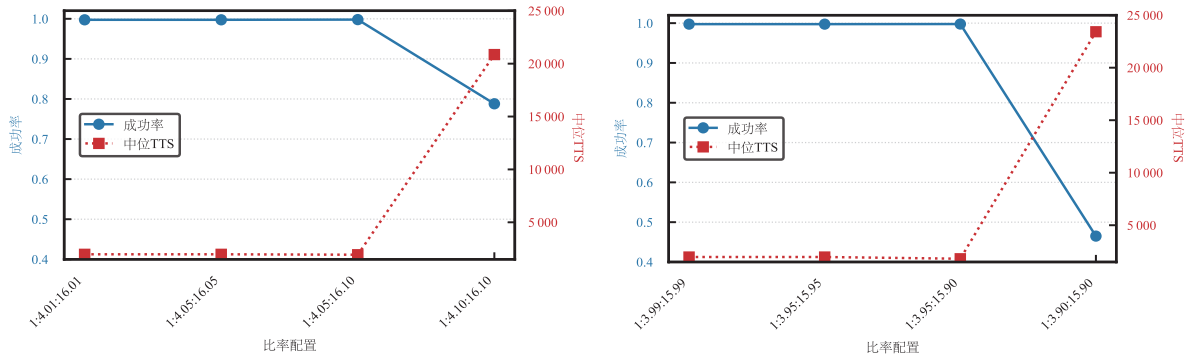


(a) SAT求解结果随器件标准差的变化情况

(b) SAT求解结果在反向传播阵列随不同电阻状态比例的变化情况

(a) SAT solving results under different device deviations

(b) SAT solving results under different resistance ratios in the backward-pass array



(c) SAT求解结果在正向传播阵列随不同电阻状态比例的变化情况，其中电导比例朝偏大方向变化

(d) SAT求解结果在正向传播阵列随不同电阻状态比例的变化情况，其中电导比例朝偏小方向变化

(c) SAT solving results under different resistance ratios in the forward-pass array, when the conductance ratio shifts towards larger values

(d) SAT solving results under different resistance ratios in the forward-pass array, when the conductance ratio shifts towards smaller values

图 11 SAT求解结果随不同参数的变化情况

Figure 11 SAT solving results under different parameters

或以上精度的模数转换器。降低电阻比和读出精度虽能减少功耗,但会加剧串扰影响。图 11(b)表明,1:16电阻比能保持与原始数学结果相近的性能,仅当电阻比降至1:4时才会出现严重性能退化。反向传播阵列电阻比的选择受到计算精度、计算功耗以及相关电路面积的影响。反向传播阵列的计算精度估算主要考虑多个高阻态电流加和超过低阻态电流所带来的计算误差。考虑电阻比例达到1:8~1:16范围内,在计算精度角度,其在Fold-SAT计算中效果相近。在计算功耗角度,考虑构成功耗的主要部分包括阻变存储器中电流产生的功耗与数据读出功耗(模数转换器功耗)。本工作采用先前工作中的模数转换器及其变化趋势用于估算<sup>[28-30]</sup>。由于不同电阻比例下读出数据范围不同,本工作考虑在不同电阻比例下使用不同精度的模数转换器。当电阻比例越高,所需要的模数转换器精度也越高。在估算中以逐次逼近型(Successive Approximation Register, SAR)ADC为例,如果对于不同情况设定相同的高阻态,而低阻态大小依据电阻比例而变化,那么当电阻比例较高时(如1:16),

阵列的功耗也就越高;相应地,更低的电阻比例(如1:8)对应的阵列功耗则更低。这里的电阻比例高、低主要指高阻态相比低阻态的电阻倍数高、低。综合阵列功耗与读出电路功耗,如图 12所示,反向传播阵列功耗随着电阻比例的增大而增大。在面积方面,由于在反向传播阵列中固定同一列中存储变量 $V_i$ 及其取反 $\bar{V}_i$ 所对应的权重,不同电阻比例不影响其所占据的阵列规模大小。在用于读出的模数转换器方面,更高精度的模数转换器会带来更大的面积开销。当电阻比例越大,反向传播阵列所需要的面积也就越大,其增长来源主要为读出电路开销。其中,比例系数1:12~1:16使用了同样的模数转换器精度,因此其面积开销相近。综合而言,更高的电阻比例有助于提升计算精度,但会引入更高的功耗和面积开销。1:8的电阻比有望成为在相近精度下更节约功耗和面积的比例选择。在满足计算精度的范围内,可以选择其中较小的电阻比例。而在不同的电阻比例下,应当尽可能使用如前所述方法降低阵列误差。

Fold-SAT中多级子句存储方案的实施依赖于阻

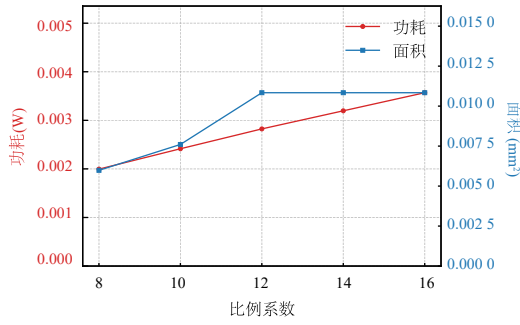


图12 反向传播阵列的功耗与面积随着不同电阻比例系数的变化

Figure 12 The power consumption and area cost of the backward-pass array under different resistance ratio

变存储器电阻的线性分布特性。本研究深入分析了电阻比非线性对系统性能的影响规律。如图11(c)所示, Fold-SAT可耐受一定程度的电阻比非线性, 当RS1中值电导相对于RS0的倍数从4.01倍波动至4.10倍, 或RS2从16.01倍波动至16.10倍时, 系统性能保持稳定。然而当RS1和RS2的电导值均朝高阻方向偏离RS0基准值的10%时, 性能会出现急剧退化。仿真结果进一步表明, 如图11(d)所示, 若电导值朝低阻方向偏离10%, 将导致成功率显著下降及中位求解时间大幅增加。

通过研究存在器件间波动的阵列表现, 本文证实Fold-SAT在实际场景中具备一定的误差容忍能力。但实现该方案仍需对阻变存储器阵列进行高精度编程及降噪处理。本研究为前期工作中提出的阻变存储器编程技术提供了潜在应用场景, 展现了其在实际系统中的适用性。

在运行时间方面, 考虑每个迭代周期消耗5个时钟周期, 并且对照原有并行梯度<sup>[13]</sup>工作设置时钟频率为500 MHz, 那么根据平均运行时间(Median Time-To-Solution, Median TTS)可以估算得到, 本工作针对 $N=20$ 的问题求解时间约为 $2.95 \mu\text{s}$ ; 针对 $N=50$ 的问题求解时间约为 $19.74 \mu\text{s}$ ; 针对 $N=100$ 问题的求解时间约为 $117.72 \mu\text{s}$ 。在相同的时钟频率设置下, 本工作的运行时间相对原有并行梯度工作有一定延迟, 但是获得了阵列规模的节约。而相比于其他已有工作, 在50变量、218子句的3-SAT问题上能够取得一定速度优势。相比基于振荡器的实现方式<sup>[31]</sup>, 求解时间只相当于其约62%, 求解时间与VIP-Sat<sup>[11]</sup>和SKADI<sup>[32]</sup>接近, 说明本工作在求解速度方面可以与当前领域内的先进水平相当, 同时节约其所需的阵列规模。

本研究中的Fold-SAT方法可以有效减少在使用并行梯度方法求解可满足性过程中所需要的阵列器件数量。需要注意, 前向传播阵列和反向传播阵列的读出电路精度需要随着使用多值级别数量的增加而

增加。因此, 一列中折叠的子句数量与读出电路精度及其带来的硬件开销成为一种取舍关系。在未来的研究中, 需要考虑这种取舍关系, 进一步研究其计算部署。

在前向传播阵列的硬件映射中, 有可能临近3个子句出现变量重合的情况。在这种情况下, 需要调整子句之间的排列顺序, 避免具有重合变量的子句出现在同一列中。此前工作曾使用启发式算法进行阵列存储优化<sup>[33]</sup>。针对这一问题, 本文考虑使用如算法1所示的启发式优化算法来优化子句之间的顺序。需要指出的是, 该方法并不包含在求解SAT问题的算法本身中。由于在一个合取表达式中, 各个子句之间的顺序并不影响求解, 本方法目的在于在初始化形成合取表达式的过程中, 使子句之间的排布更适合纳入于Fold-SAT之中。

#### 算法1 基于队列的子句分组算法

输入: 子句矩阵、子句分组大小(默认值为3)

输出: 优化后的子句矩阵

```

1.   WHILE 队列不为空 DO
2.       取出队首子句
3.       # 检查是否可以加入当前组:
4.       IF 当前组为空 THEN
5.           直接加入
6.       ELSE
7.           IF 与组内所有子句无重叠 THEN
8.               成功加入
9.           ELSE
10.              重新入队
11.              IF 强制加入条件满足 THEN
12.                  强制加入组
13.              END IF
14.          END IF
15.      END IF
16.      #检查步骤结束
17.      IF 可以加入 THEN
18.          加入当前组
19.      ELSE
20.          重新入队
21.          IF 尝试次数超过限制 THEN
22.              强制加入当前组
23.          END IF
24.      END IF
25.      IF 当前组大小 = 子句分组大小 THEN
26.          保存当前组
27.          初始化新组
28.      END IF
29.  END WHILE

```

算法 1 展示了一种基于队列的子句分组算法。假设给定一个 SAT 问题中未经优化的合取表达式, 目标为将其中每 3 个子句分为一组, 对应 Fold-SAT 中存储在一列的 3 个子句, 要求尽可能少地出现组内变量重合。首先将合取表达式中的子句排成一个队列, 按照队列顺序逐个纳入分组。对于每个子句, 检查是否可以加入当前组: 如果当前组为空, 则直接加入; 如果当前组内存在子句, 则与子句比较有无重叠。如果该子句与当前组存在重叠, 则将当前组排入队尾, 并考虑队列中下一个子句。设定尝试次数限制, 如果超过次数限制而没有找到合适的子句, 则将当前子句加入。针对于存在变量重叠的组, 在映射至 Fold-SAT 阵列时, 把组内变量单独存储于不同列, 用于避免映射存在重叠。算法结束后, 对于没有完全消除重叠的合取表达式, 会在映射至 Fold-SAT 时使用一些额外的列来避免重叠。

在 SATLIB 数据集上进行分析, 多数合取表达式在使用算法 1 优化后能够避免重叠, 少数存在重叠的情况一般只有 1~3 个重叠组, 所需要增加的列数占总列数比例低。对于子句重叠的检查, 同样可以使用存内计算或近存计算的方式实现, 一个计算周期中可以检查一个组是否存在重叠。考虑到这种实现方式, 对于具有  $N$  个子句的问题, 每个子句中有  $K$  个变量, 那么该算法的时间复杂度在  $O(N)$  (最好情况) 到  $O(N^2)$  之间。在 SATLIB 数据集中的问题上进行这一算法仿真, 考虑一个存内计算时间周期为 10 ns, 可以估算得到对于给定变量的 SAT 问题所需要的初始化分组时间。基于以上分析, 在数据集上的平均分组时间约为 4  $\mu$ s (50 变量情况)、7  $\mu$ s (100 变量情况)。在更高的时钟频率下运行时间还可以进一步节约。由于分组算法是在存储子句之前进行的一种初始化步骤, 这一延时针对于特定场景, 需要在计算中一起完成初始化与 SAT 求解时才会一起使用。而如果是提前存储好 SAT 问题, 只有在需要使用时调用, 那么进行求解时只需要调用 Fold-SAT 本身即可。

### 3 结论

在硬件布尔可满足性求解器中, 针对于存内计算阵列映射子句所面临的稀疏性问题, 本研究提出了一种利用存内计算阵列中多值阻变存储器与子句折叠策略的并行梯度布尔可满足性求解器。该方法压缩了并行梯度计算中所需要的存储器规模, 将前向传播阵列所需的存储器件数量压缩至传统方案的 1/3, 将反向传播阵列的存储器件数量压缩至传统方案的 1/2。前向传播阵列可采取 1:4:16 的电阻态比例, 而反向传播阵列的电阻态比例在 1:8~1:16 之间, 具有一定

选择范围。本研究还探索了器件非理想效应带来的影响。仿真结果表明, 器件偏差范围需要在 1  $\mu$ S 以内, 而电导均值相对于基准值的偏移应控制在 10% 以内, 更精确的编程有助于提升求解过程的准确性。

### 参考文献

- [1] Biere A, Heule M, van Maaren H, et al. Handbook of satisfiability: Part 1/Part 2[M]. 2nd ed. Amsterdam: IOS Press, 2021.
- [2] Schaefer T J. The complexity of satisfiability problems[C]// Proceedings of the tenth annual ACM symposium on theory of computing. San Diego: ACM, 1978: 216-226.
- [3] Marques-Silva J. Practical applications of Boolean satisfiability[C]//2008 9th international workshop on discrete event systems. Gothenburg: IEEE, 2008: 74-80.
- [4] Ganai M K, Gupta A. SAT-based scalable formal verification solutions[M]. New York: Springer, 2007.
- [5] Prasad M R, Biere A, Gupta A. A survey of recent advances in SAT-based formal verification[J]. International Journal on Software Tools for Technology Transfer, 2005, 7(2): 156-173.
- [6] Marques-Silva J P, Sakallah K A. Boolean satisfiability in electronic design automation[C]//Proceedings of the 37th annual design automation conference. Los Angeles: ACM, 2000: 675-680.
- [7] Li Min, Shi Zhengyuan, Lai Qiuxia, et al. On EDA-driven learning for SAT solving[C]//2023 60th ACM/IEEE design automation conference. San Francisco: IEEE, 2023: 1-6.
- [8] Martins R, Manquinho V, Lynce I. An overview of parallel SAT solving[J]. Constraints, 2012, 17(3): 304-347.
- [9] Goldberg E, Novikov Y. BerkMin: A fast and robust sat-solver[J]. Discrete Applied Mathematics, 2007, 155(12): 1549-1561.
- [10] Xie Shanshan, Yang Mengtian, Lanham S A, et al. 29.2 snap-SAT: A one-shot energy-performance-aware all-digital compute-in-memory solver for large-scale hard Boolean satisfiability problems[C]//2023 IEEE international solid-state circuits conference. San Francisco: IEEE, 2023: 420-422.
- [11] Shim C, Bae J, Kim B. 30.3 VIP-Sat: A Boolean satisfiability solver featuring 5 $\times$ 12 variable in-memory processing elements with 98% solvability for 50-variables 218-clauses 3-SAT problems[C]//2024 IEEE international solid-state circuits conference. San Francisco: IEEE, 2024: 486-488.
- [12] Kim D, Rahman N M, Mukhopadhyay S. 29.1 a 32.5 mW

- mixed-signal processing-in-memory-based k-SAT solver in 65 nm CMOS with 74.0% solvability for 30-variable 126-clause 3-SAT problems[C]//2023 IEEE international solid-state circuits conference. San Francisco: IEEE, 2023: 28-30.
- [13] Bhattacharya T, Hutchinson G H, Pedretti G, et al. Computing high-degree polynomial gradients in memory[J]. *Nature Communications*, 2024, 15(1): 8211.
- [14] Sohahngpurwala A A, Hassan M W, Athanas P. Hardware accelerated SAT solvers: A survey[J]. *Journal of Parallel and Distributed Computing*, 2017, 106: 170-184.
- [15] Ivan T, Aboulhamid E M. An efficient hardware implementation of a SAT problem solver on FPGA[C]//2013 Euromicro conference on digital system design. Los Alamitos: IEEE, 2013: 209-216.
- [16] Nguyen A H N, Aono M, Hara-Azumi Y. FPGA-based hardware/software co-design of a bio-inspired SAT solver[J]. *IEEE Access*, 2020, 8: 49053-49065.
- [17] Zhang Qiaochu, Su Shiyu, Liu Zerui, et al. A stochastic analog SAT solver in 65 nm CMOS achieving 6.6  $\mu$ s average solution time with 100% solvability for hard 3-SAT problems[C]//2024 IEEE symposium on VLSI technology and circuits (VLSI technology and circuits). Honolulu: IEEE, 2024: 1-2.
- [18] Yan Bonan, Hsu J L, Yu P C, et al. A 1.041-mb/mm<sup>2</sup> 27.38-TOPS/W signed-INT8 dynamic-logic-based ADC-less SRAM compute-in-memory macro in 28 nm with reconfigurable bitwise operation for AI and embedded applications[C]//2022 IEEE international solid-state circuits conference. San Francisco: IEEE, 2022: 188-190.
- [19] Verma N, Jia Hongyang, Valavi H, et al. In-memory computing: Advances and prospects[J]. *IEEE Solid-State Circuits Magazine*, 2019, 11(3): 43-55.
- [20] Wan Weier, Kubendran R, Schaefer C, et al. A compute-in-memory chip based on resistive random-access memory[J]. *Nature*, 2022, 608(7923): 504-512.
- [21] 李嘉宁, 姚鹏, 揭路, 等. 存算一体技术研究现状[J]. *电子学报*, 2024, 52(4): 1103-1117.  
Li Jianing, Yao Peng, Lu Jie, et al. Research status of computing-in-memory technology[J]. *Acta Electronica Sinica*, 2024, 52(4): 1103-1117. (in Chinese)
- [22] Li Wantong, Sun Xiaoyu, Huang Shanshi, et al. A 40-nm MLC-RRAM compute-in-memory macro with sparsity control, on-chip write-verify, and temperature-independent ADC references[J]. *IEEE Journal of Solid-State Circuits*, 2022, 57(9): 2868-2877.
- [23] Park J, Kumar A, Zhou Yucheng, et al. Multi-level, forming and filament free, bulk switching trilayer RRAM for neuromorphic computing at the edge[J]. *Nature Communications*, 2024, 15(1): 3492.
- [24] Wu Wei, Wu Huaqiang, Gao Bin, et al. A methodology to improve linearity of analog RRAM for neuromorphic computing[C]//2018 IEEE symposium on VLSI technology. Honolulu: IEEE, 2018: 103-104.
- [25] Du Yiwei, Tang Jianshi, Li Yijun, et al. Monolithic 3D integration of analog RRAM-based computing-in-memory and sensor for energy-efficient near-sensor computing[J]. *Advanced Materials*, 2024, 36(22): 2302658.
- [26] Song Wenhao, Rao Mingyi, Li Yunning, et al. Programming memristor arrays with arbitrarily high precision for analog computing[J]. *Science*, 2024, 383(6685): 903-910.
- [27] Rao Mingyi, Tang Hao, Wu Jiangbin, et al. Thousands of conductance levels in memristors integrated on CMOS[J]. *Nature*, 2023, 615(7954): 823-829.
- [28] Jo Y J, Yap B P, Yoon D H, et al. DenseCIM: Binary weighted-capacitor SRAM computation-In-memory with column-by-column dynamic range calibration SAR ADC[C]//2023 IEEE custom integrated circuits conference. San Antonio: IEEE, 2023: 1-2.
- [29] Yip M, Chandrakasan A P. A resolution-reconfigurable 5-to-10-bit 0.4-to-1 V power scalable SAR ADC for sensor applications[J]. *IEEE Journal of Solid-State Circuits*, 2013, 48(6): 1453-1464.
- [30] 郭仲杰, 苏昌勛, 许睿明, 等. 基于粗细量化并行与TDC混合的CMOS图像传感器列级ADC设计方法[J]. *电子学报*, 2024, 52(2): 486-499.  
Guo Zhongjie, Su Changxu, Xu Ruiming, et al. Column level ADC design method of CMOS image sensor based on coarse and fine quantization parallel and TDC hybrid[J]. *Acta Electronica Sinica*, 2024, 52(2): 486-499. (in Chinese)
- [31] Dikopoulos E, Hsu Y T, Wormald L, et al. 25.1 a physics-inspired oscillator-based mixed-signal optimization engine for solving 50-variable 218-clause 3-SAT problems with 100% solvability and 31.7  $\mu$ s solution time[C]//2025 IEEE international solid-state circuits conference. San Francisco: IEEE, 2025: 1-3.
- [32] Wu Zihan, Tang Xiyuan, Zhang Tao, et al. SKADI: A 28-nm complete K-SAT solver featuring bidirectional in-memory deduction and incremental updating[J]. *IEEE Journal of Solid-State Circuits*, 2025: 1-14.
- [33] Yan Bonan, Liu Mengyun, Chen Yiran, et al. On designing efficient and reliable nonvolatile memory-based com-

puting-In-memory accelerators[C]//2019 IEEE international electron devices meeting. San Francisco: IEEE,

2020: 14.5.1-14.5.4.

### 作者简介



**岳文硕** 男,1999年1月出生于北京市。2021年于北京大学获得学士学位。现为北京大学集成电路学院博士研究生。主要研究方向为忆阻器存内计算。

E-mail: yws2017@pku.edu.cn



**赵洪霄** 男,2001年12月出生于江苏省宿迁市。现为北京大学人工智能研究院博士研究生。主要研究方向为认知架构及其应用。

E-mail: hongxiaozhao@stu.pku.edu.cn



**叶林韬** 男,2003年8月出生于北京市。2025年毕业于北京大学信息科学技术学院。现为美国南加州大学博士研究生。主要研究方向为新型存储芯片。

E-mail: lintao@pku.org.cn



**李嘉怡** 女,2001年3月出生于山西省运城市。现为北京大学人工智能研究院博士研究生。主要研究方向为强化学习的硬件加速及片间互联系统。

E-mail: lijiaiyi@stu.pku.edu.cn



**史代璟** 男,1999年12月出生于福建省三明市。现为北京大学集成电路学院博士研究生。主要研究方向为存内计算外围电路设计。

E-mail: daijingshi@pku.edu.cn



**张腾** 男,1994年10月出生于安徽省芜湖市。现为北京大学助理研究员。主要研究方向为后摩尔器件集成、感存算一体。

E-mail: tengzhang@pku.edu.cn



**符一涵** 男,2000年10月出生于江苏省丹阳市。2023年于北京大学获得学士学位。现为北京大学人工智能研究院博士研究生。主要研究方向为计算机体系结构与人工智能加速器。

E-mail: yihanfu@pku.edu.cn



**燕博南** 男,1991年6月出生于河北省保定市。现为北京大学人工智能研究院助理教授,北京大学小米博雅青年学者,博士生导师,北京大学通用人工智能芯片研究中心执行主任。主要研究方向为通用人工智能体处理器芯片设计、SRAM与新兴存储器(MRAM、RRAM、PCM、FeFET等)集成电路设计。

E-mail: bonan@pku.edu.cn



**杨宇翔** 男,2001年2月出生于广东省梅州市。现为北京大学集成电路学院博士研究生。主要研究方向为忆阻器及其应用。

E-mail: yangyuxiang@pku.edu.cn



**杨玉超** 男,1984年12月出生于河北省衡水市。现为北京大学博雅特聘教授,国家杰青,新基石研究员,信息工程学院院长,科学智能学院副院长。主要研究方向为存算一体芯片、类脑计算、脑机接口等。中国电子学会会员编号: E190014285M。

E-mail: yuchaoyang@pku.edu.cn