

基于存储器编译器的敏捷生成技术研究

刘孙辰星, 蔡浩*

(东南大学 集成电路学院, 南京 211189)

摘要: 磁随机存储器作为一种新型非易失性存储, 因其优良的读写速度与耐久度特性, 在嵌入式存储领域具有广阔的应用前景。然而, 由于磁随机存储器的定制化设计通常需要数月完成, 具有较长的设计周期, 这与片上系统较快的设计迭代需求存在一定矛盾。存储器编译器作为一种快速生成存储器设计的工具, 是解决这一矛盾的有效手段。本文从磁随机存储器的全定制设计流程出发, 同时对各类存储器编译器的研究现状开展调研, 总结了目前存储器编译器工作的现状与挑战, 最终讨论了磁随机存储器编译器的设计方法学。

关键词: 存储器编译器; 磁随机存储器; 电子设计自动化; 敏捷设计

中图分类号: TP872 **文献标识码:** A

Research for agile development technique based on memory compiler

LIU Sunchenxing, CAI Hao

(School of Integrated Circuits, Southeast University, Nanjing 211189, China)

Abstract: As novel non-volatile memory, magneto-resistive random access memory (MRAM) has broad application prospects in the field of embedded memory due to its excellent read and write speed and endurance characteristics. However, since the customized design of MRAM usually takes several months to complete, it requires a long design period, which conflicts with the need for a faster design iteration of the system-on-chip. As a tool for quickly generating memory designs, the memory compiler is an effective means to resolve this contradiction. This article starts with the fully customized design process of magnetic random access memory and conducts a survey on the research status of various types of memory compilers. We summarize the current status and challenges of memory compiler design from published work and finally discuss the design methodology of magnetic random access memory compiler.

Keywords: magneto-resistive random access memory; memory compiler; electronic design automation; agile development

0 引言

磁随机存储器 (Magneto - Resistive Random Access Memory, MRAM) 作为一种新型非易失性存储, 通过磁性材料的磁性状态完成数据的存储。相比传统的存储器, MRAM 具有非易失特性, 结合静态随机存储器 (Static Random Access Memory, SRAM) 的高读写速度以及动态随机存储器 (Dynamic Random Access Memory, DRAM) 的高密度特征, 在读写速度和耐久度方面具有显著优势。因此, MRAM 在嵌入式存储器应用领域具有可观的应用前景^[1]。然而, MRAM 的定制设计具有较长的设计周期, 如何在设计需求下高效且快速地完成存储器设计的敏捷生成, 逐渐成为一项亟需解决的关键问题。由于存储器结

构的规则性, 为存储器的自动化生成提供了良好的基础。存储器编译器作为高效完成存储器敏捷设计生成的一种技术, 能够根据用户定义的设计参数自动化地完成设计文件的编译生成, 生成的设计文件包括电路网表与版图文件等。在这一背景下, MRAM 的编译生成是针对全定制 MRAM 的设计周期长与片上系统 (System on Chip, SoC) 中存储器设计快速迭代需求之间矛盾的有效解决方法。

本文从 MRAM 全定制设计的基本流程与 MRAM 关键电路结构出发, 结合各类传统与新兴存储器编译器的研究现状, 探讨了 MRAM 编译器设计方法。

1 MRAM 全定制设计流程

一般来说, MRAM 采用全定制的设计流程, 具有较长

基金项目: 国家自然科学基金面上项目 (62274029)。

刘孙辰星 (硕士研究生), 主要研究方向为磁随机存储器电路设计、数字集成电路设计。

* 通信作者。 E-mail: hao.cai@seu.edu.cn。

的设计周期。全定制 MRAM 存储器的设计流程指的是从设计要求出发,考虑存储容量、读写速度、功耗与良率等设计指标,从最小尺寸的元器件 MTJ 的设计开始,人工定制地完成 MRAM 的完整设计。

如图 1 所示,设计流程包括前端电路设计以及后端设计和表征两个部分。在前端设计中,需根据设计需求定制 MRAM 位单元并拓展至一定规模的存储阵列。此外,还需对外围电路进行定制设计,包括地址译码电路、读写驱动模块以及读灵敏放大器等关键电路模块。MRAM 存储阵列与外围电路定制完成后,得到完整的 MRAM 宏单元前端电路设计。前端电路设计功能验证正确后,根据工艺规定的设计规则绘制电路版图,完成 MRAM 宏单元的版图物理实现。经过设计规则检查等版图验证环节,抽取版图寄生信息以开展版图后仿真工作。对于特定的设计来说,版图后仿真工作需要在不同的工艺角、输入压摆以及输出负载下开展,以完成延时与功耗的特征化提取。经过版图后仿真得到的时序库与标准延时文件,结合设计需求用以指导前端电路设计的优化工作。电路设计与反馈优化过程相结合,以得到符合设计需求的 MRAM 宏单元定制设计。

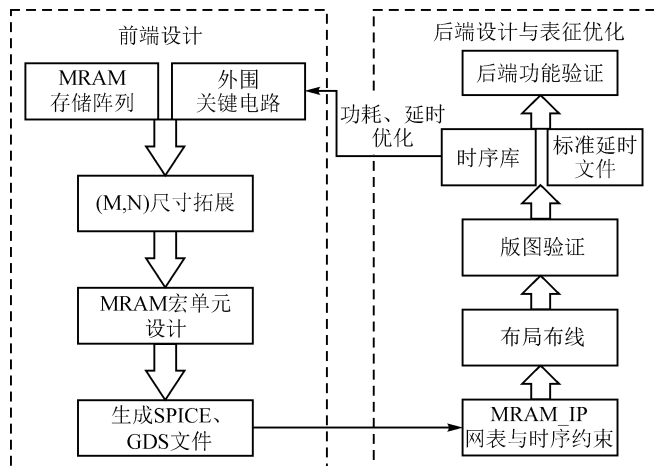


图 1 全定制 MRAM 设计流程

Fig. 1 Customized MRAM design flow

这种全定制的设计流程,其周期长、设计方法复杂,与 SoC 设计的快速迭代需求存在一定矛盾。在此背景下,存储器编译器可以成为一种有效的解决方案,加速 SoC 中存储器的设计周期,实现设计的快速迭代。

2 磁随机存储器关键模块设计概述

MRAM 宏单元设计的结构框图如图 2 所示,主要包括存储单元阵列以及译码器、驱动电路、灵敏放大器等关键电路模块。译码电路根据输入地址寻址到指定的存储单元,驱动电路提供读写操作时加载在存储阵列的电压,

灵敏放大器是读操作的关键电路模块。

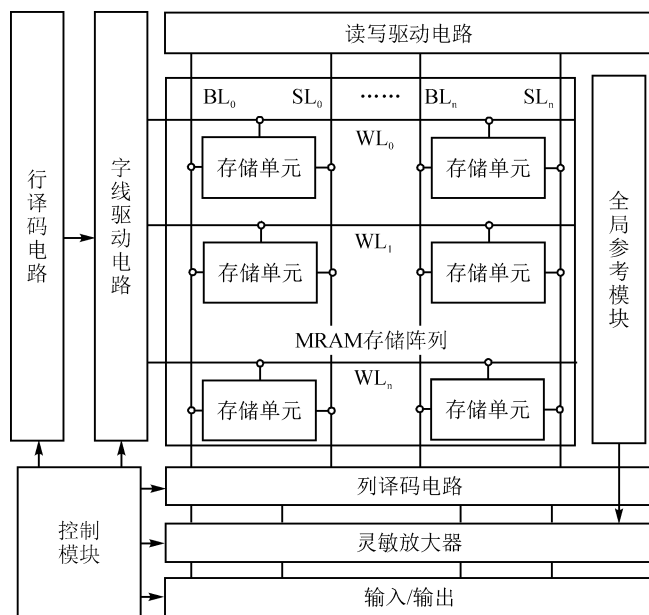


图 2 MRAM 宏单元结构框图

Fig. 2 MRAM macro design diagram

在存储器编译过程中,根据每个电路模块设计需求的不同,将它们分为预先提供的库单元模块以及动态生成电路模块。其中,“硬模块”是宏单元中需要手工定制设计的模块,以获得更好的性能表现。在 MRAM 宏单元中,灵敏放大器作为需要精确设计的模拟电路,为了得到更好的读良率与读速度,需要进行手工定制。写驱动电路需要根据位线上的电容负载情况手动调整驱动器晶体管尺寸,因此作为定制的库单元模块。灵敏放大器与驱动模块在编译生成时按照位宽对应调用生成。对于动态生成模块,需要预先提供可指定类型和大小的晶体管,作为电路模块动态生成的基础。在 MRAM 电路模块中,对于行、列译码电路等模块进行动态生成。存储阵列作为重复出现的主要电路结构,是所占面积比重最大的部分,一般来说需要对位单元的结构定制化完成设计,以获得最好的面积效益与优化的性能表现。在不同的尺寸配置下,相应地进行拼接。

3 存储器编译器研究进展

3.1 存储器编译器概述

存储器编译器的主要功能在于生成存储器的网表文件等标准文件,表征时序功耗信息填入库文件、完成存储器版图的物理实现与验证等。

存储器编译器的工作流程主要分为两个阶段。第一阶段完成存储器设计文件的编译生成,设计文件包括门级网表、版图文件以及 Verilog 行为模型。第二阶段对生成

的存储器设计进行性能表征,以时序、功耗等参数体现。

图3描述了存储器编译器的一种通用设计流程。编译器具有在工艺节点间可迁移的特征,对于每一个编译器支持的工艺节点,需要预先提供基本单元库,包括基本电路模块库与基本版图库,作为设计文件编译生成的基础。因此,编译器运行拼接算法之前,需要对原理图模块和版图子模块进行划分,根据划分的单元电路生成基本单元库,编译器才能调用这些基本单元库与算法进行拼接。

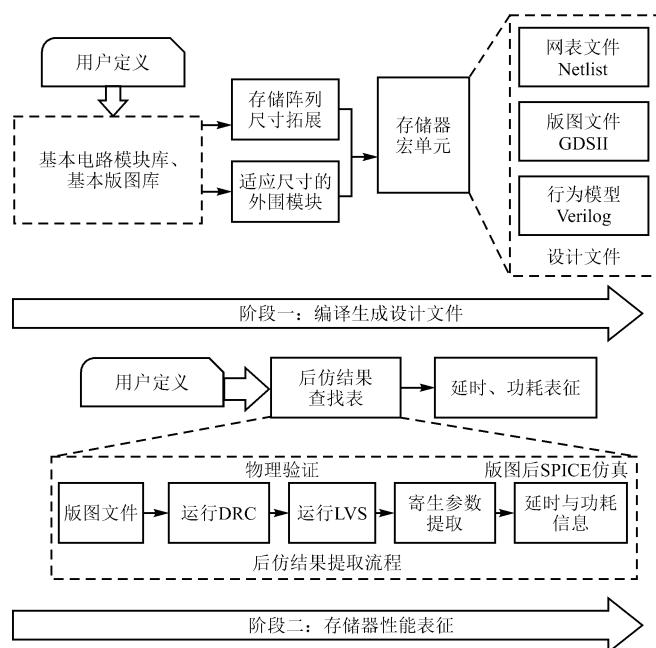


图3 一般的编译器工作流程

Fig. 3 General work flow of memory compiler

首先,编译器完成设计文件的自动编译生成。用户根据其设计需求定义编译器的基本参数,可供用户定义的基本参数类型如表1所列。编译器调用预先提供的基本单元库,将预先提供的存储阵列拓展至目标尺寸的阵列,并对外围电路模块进行尺寸拓展,层次化地完成存储器宏单元设计文件的编译生成,包括网表文件、版图文件以及行为模型等标准文件。

表1 用户可定义的基本参数表

Table 1 Basic parameters for user define

用户定义变量类型	变量含义
Words (W)	编译器存储深度
Bits (B)	编译器数据位宽
CMUX (N)	列选择器宽度

此后,对编译生成的存储器设计进行性能表征,主要针对延时参数以及性能参数。对于一般的存储器编译器,设计公司会提供有限的参数配置下的仿真数据,供用户参考其定义配置下的性能表现情况。

假定一个存储容量极限为 1024×256 的编译器,若限定可选择的 CMUX 宽度为 4,存储深度步长为 16,存储字宽步长为 4,存储深度可选范围为 $64 \sim 1024$,存储位宽可选范围为 $8 \sim 256$,则所需的仿真数为 3 720。提供可选范围内所有设计的仿真数据,供存储器编译器为用户生成性能表征预览是不实际的,所需仿真时间成本很高。因此,高效且可靠的存储器性能表征建模或预测方法的提出对于存储器编译器来说是必要的。

存储器编译器在工业界已广泛投入使用,如台积电的 55 nm 工艺 SRAM 编译器,但其内部的设计文件编译过程往往不可见。此外,新思科技 (Synopsys) 公司在 2014 年发表了一项通用存储器编译器工作,能够在多种抽象工艺下生成 SPICE 网表、GDSII 版图信息、Verilog 行为模型以及时序功耗库文件等。然而,这项通用存储器编译器工作只能用于模拟存储器的设计流程,因为它不支持可供制造的工艺库^[2]。此外,学术界也在存储器编译器领域公开了相关工作,东南大学、清华大学以及中国科学院微电子研究所等单位提出了 SRAM 编译器的方法学与设计流程,但都没有公开发表的完整编译器工作^[3-5]。加州大学圣克鲁兹分校首先提出了开源的存储器编译器开发框架 OpenRAM,并实现了多个工艺节点下的可迁移^[6]。

3.2 SRAM 编译器研究进展

静态随机存储器作为一种传统的存储器件,具有较为标准的传统设计方法,且由于 SRAM 位单元由晶体管构成,使 SRAM 天然具有兼容标准 CMOS 工艺的特点,为 SRAM 的自动编译生成自动化提供了便利,且易于在工艺节点间迁移。因而,现有的存储器编译器设计,尤其是工业存储器编译器,一般始于面向 SRAM 的编译工作。

美国新思科技在 2014 年发表了一项通用存储器编译器软件工具。所提出的通用存储器编译器支持 90 nm 与 32/28 nm 的抽象工艺,生成的设计只用于存储器设计流程的教育目的,所支持的抽象工艺不可制造。生成的设计文件包括 SPICE 网表、GDSII 版图信息、Verilog 行为模型以及时序功耗库文件等。

加州大学圣克鲁兹分校在 2014 年提出了一种开源的存储器编译器框架,支持可制造的工艺,同时易于在工艺节点间实现迁移^[6]。提出的编译器工作框架如图4所示,分为“前端”和“后端”两种模式。前端模式分为编译器和表征器两部分,基于用户定义的设计需求完成前端设计文件的敏捷生成与初步特征化。编译器生成存储器 SPICE 模型以及 GDSII 格式的版图信息等标准设计文件,表征器调用 SPICE 仿真器生成前端初步的时序与功耗结果。后端模式下表征器调用 SPICE 仿真器得到准确的时序与功耗结果,因为后端模式下的仿真基于 GDSII 版图抽取

的 SPICE 网表,计入了存储器的版图寄生情况。此外,还提出存储器编译器的可改进方向,增加可配置的 SRAM 不同设计、加速特征化方法以及增加 GUI 界面等。

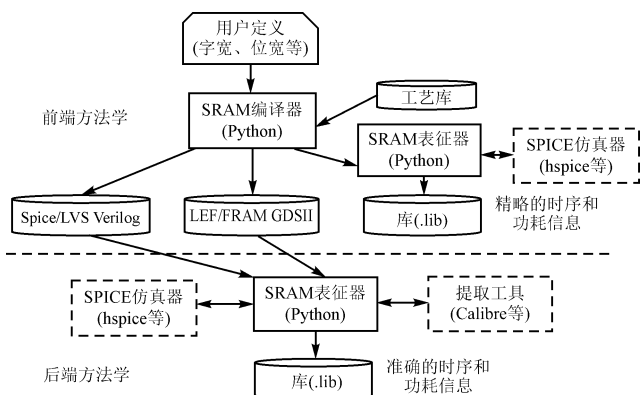


图 4 OpenRAM 提出的整体编译与特征化方法

Fig. 4 Overall compile and characterize methodology of OpenRAM

东南大学、清华大学等单位也在存储器编译器领域发表了嵌入式 SRAM 编译器的方法学与流程设计。在 2007 年与 2010 年,清华大学先后发表了 SRAM IP 编译器工作,实现了半自动的 SRAM IP 编译生成^[3-4]。对于存储器性能特征化,采用了基于仿真数据的查找表方法。此后,结合外推法对仿真容量范围之外的存储器时序信息做出预测,平均误差在 5% 左右。在 2012 年,东南大学发表的 SRAM 编译器工作中重点关注了基于重叠距离的版图布局方案,以及分段多项线性插值建模的时序、功耗预测方法,所提出的编译器生成的 SRAM IP 设计经过中芯国际 65 nm 工艺流片验证^[5]。

在 2023 年,香港科技大学发表了 SRAM 存算架构的编译器工作 AutoDCIM,是首个面向数字存内计算的自动化编译工作^[7]。如图 5 所示,AutoDCIM 提出了一种基于硬件约束的设计优化方法,在面积、延时与能效设计需求下搜索存算阵列的布局方法,得到最优化的设计方案。将自动编译生成的存算宏单元与近年发表的设计对比^[8-9],面积效率达到平均水平,而在能效指标上有着显著提升。

3.3 RRAM 编译器研究进展

RRAM 与 MRAM 同样作为新型非易失存储,RRAM 的设计流程与 MRAM 相似。近年来,学术界在存储架构与存算架构下的 RRAM 编译器工作做出了积极尝试,对 MRAM 编译器的设计具有重要的参考价值。

在 2021 年和 2022 年,帝国理工大学分别发表了 RRAM 编译器的开源工作^[10-11],前一项工作中实现了 RRAM 阵列生成与版图验证,在后一项工作中完善了编译器工作流程。如图 6 所示,编译器提供 SKILL 脚本实现基于 EDA 工具的编译自动化,在前端生成设计文件过程中引入楷登电子(Cadence)的 Virtuoso 工具,在版图验

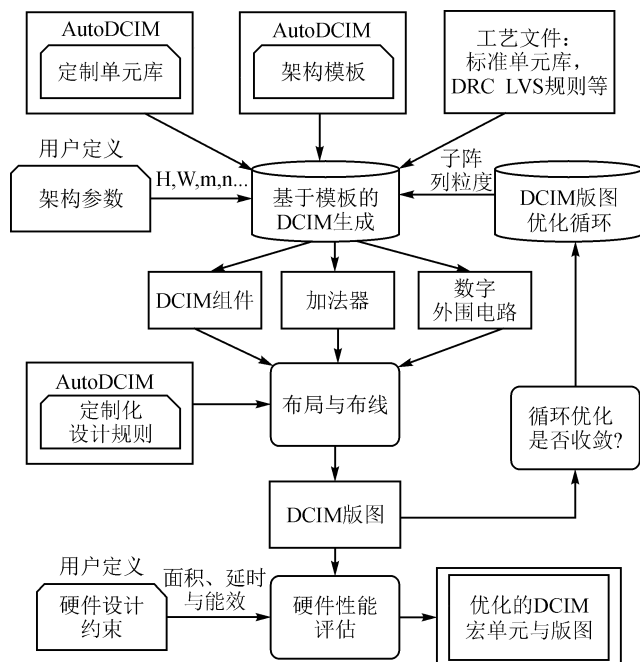


图 5 AutoDCIM 的编译工作框架

Fig. 5 AutoDCIM compilation framework

证阶段引入明导公司(Mentor)的 Calibre 工具,全流程对 EDA 工具依赖性高。提出的开源 RRAM 编译器能够实现 RRAM 系统级的自动编译生成,版图布局与验证并完成时序表征,实现了 RRAM 编译器前端到后端的完整流程,编译器功能的实现情况如表 2 所列。在先前发表的工作仅实现 RRAM 阵列生成与版图验证的基础上,提出的完整编译器工作已实现存储阵列、外围电路以及系统级的设计文件生成与版图验证,包括系统级特征提取。

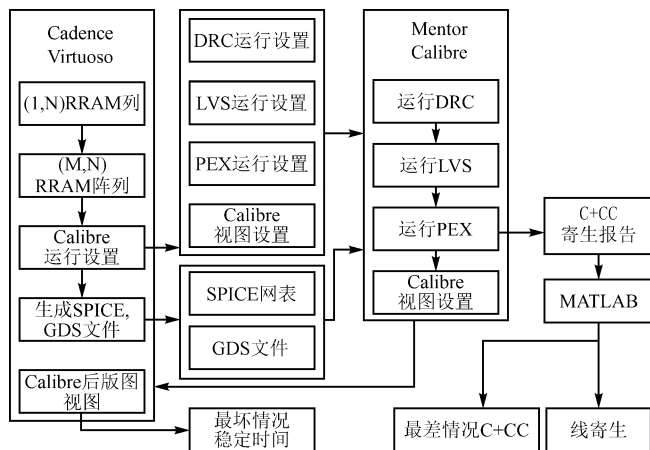


图 6 RRAM 编译器工作流程框图

Fig. 6 Workflow of RRAM compiler

安徽大学在 2022 年发表了基于 RRAM 的存内计算架构的宏单元编译生成工作^[12],比起常规的编译生成存储模块的工作,针对存算架构的编译器在设计上具有更高的灵活性。在存算结构的 RRAM 宏单元设计流程中,当

表 2 RRAM 编译器功能^{[11][12]}
Table 2 Function of RRAM compiler

结构层次	功能	参考 文献[11]	参考 文献[12]
RRAM 阵列	RRAM 阵列生成	✓	✓
	参考单元生成	×	✓
	多路选择器生成	×	✓
	写放大器生成	×	✓
	模拟 P&R	×	✓
外围模拟电路	模拟集成	×	✓
	综合	×	✓
	物理实现	×	✓
	设计约束生成	×	✓
外围数字电路	GDS 和 Verilog 引入	×	✓
	混合信号 P&R	×	✓
	混合信号集成	×	✓
系统级	版图生成	×	✓
	版图验证	✓	✓
	系统级特征化	×	✓

计算模块的设计需要做出修改时,通常需要手动对网表与版图进行重新设计,较长的设计周期不利于设计的快速迭代。提出的 RRAM 存算编译器能够基于模板文件与预先提供的库文件基于算法生成网表与 GDS 版图文件。在网表生成过程中调用逆波兰网表连接算法,在版图生成过程中采用抽象模型算法、阵列生成算法与布线算法等。这一 RRAM 存算编译器具有较好的灵活性与可配置性。RRAM CIM 编译器工作流程框图如图 7 所示。

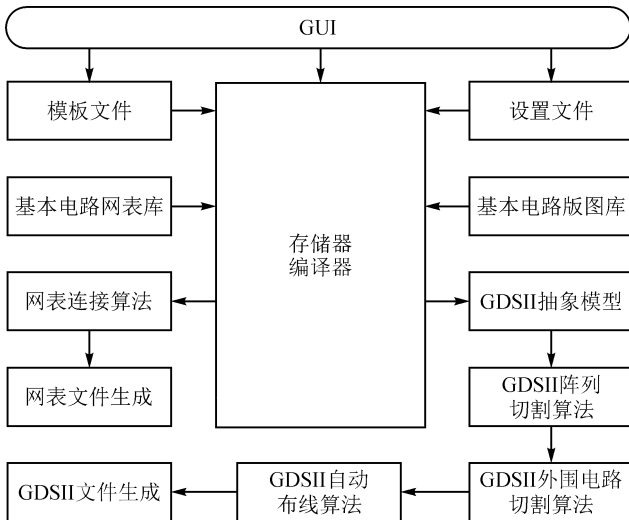


图 7 RRAM CIM 编译器工作流程框图
Fig. 7 RRAM CIM compiler workflow

3.4 MRAM 编译器研究进展

在 2022 年,三星发表了一项基于编译器生成的

MRAM 工作,在已公开的工作中达到了能效最佳表现。采用的 MRAM 编译器特征在于高密度与高能效,得到的宏单元设计达到了 12 Mb/mm² 的宏单元密度以及 25 pJ/bit 的低写入能耗^[13]。

采用的工业 MRAM 编译器适用于高密度宏单元的架构设计与表征,这项基于 28 nm FD-SOI 工艺 MRAM 编译器工作于 2020 年发表^[14]。为了在更小尺寸下开展设计验证与特征参数提取,将 128 Mb 的大容量设计划分为多个 8 Mb 亚宏单元,如图 8 所示。每一个小尺寸宏单元都具有完整的外围电路且能够独立工作,通过增加一级块译码实现 8 Mb 到 128 Mb 的尺寸拓展。提出的加速特征化方法具有层次化的特点,以 8 Mb 亚宏单元为单位进行设计验证与特征提取,仿真结果以黑盒形式供顶层特征化调用。

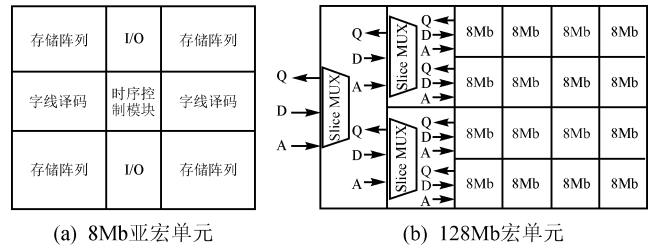


图 8 适于编译生成的 MRAM 宏单元架构设计

Fig. 8 MRAM macro design considering compiler needs

4 存储器编译器研究现状与挑战

工业界已普遍使用 SRAM 编译工具实现 SoC 设计中存储器设计的快速迭代。此外,近年有 MRAM 新型存储器编译生成的芯片工作发表。学术界也有针对存储器编译器的积极尝试,从 SRAM 到 MRAM、RRAM 等新型存储器,包括 SRAM 与 RRAM 在存算架构宏单元的自动化编译工作。

从当前存储器编译器研究工作的现状出发,分析存在以下问题与挑战:

- ① 存储器编译器目前在工业界的发展较成熟,而且编译生成的设计具有更高的可靠性,已广泛投入使用。然而,工业编译器设计可靠性的代价是较高的设计与仿真投入。
- ② 学术界发表的工作以编译器设计方法学为主,缺乏完整、开源的编译器工作,尤其对于新型存储器件更加明显。
- ③ 对于学术界的编译器工作,探索高效且准确的性能表征方案是必要的。对于编译生成的存储器,需要在不同设计特性、不同存储容量以及不同工作条件下实现快速的特征化提取。

参考文献

- [1] SALAHUDDIN S, NI K, DATTA S. The era of hyper-scaling in electronics[J]. Nature Electronics, 2018, 1(8): 442-450.
- [2] GOLDMAN R, BARTLESON K, WOOD T, et al. Synopsys'

- educational generic memory compiler [C]//10th European workshop on microelectronics education (EWME). IEEE, 2014:89-92.
- [3] Y XU, Z GAO, X HE. A flexible embedded SRAM IP compiler[C]//ISCAS, pages 3756-3759, May 2007.
- [4] S WU, X ZHENG, Z GAO, et al. A 65nm embedded low power SRAM compiler [C]//DDECS, pages 123-124, April 2010.
- [5] C MING, B NA. An efficient and flexible embedded memory IP compiler[C]//Cyber C, pages 268-273, Oct 2012.
- [6] GUTHAUS M R, STINE J E, ATAEI S, et al. OpenRAM: An open-source memory compiler[C]//2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). IEEE, 2016:1-6.
- [7] CHEN J, TU F, SHAO K, et al. AutoDCIM: An Automated Digital CIM Compiler [C]//2023 60th ACM/IEEE Design Automation Conference (DAC). IEEE, 2023:1-6.
- [8] YU - DER CHIH, PO - HAO LEE, HIDEHIRO FUJIWARA, et al. An 89 tops/w and 16.3 tops/mm² all-digital sram-based full-precision compute-in-memory macro in 22nm for machine-learning edge applications [C]//ISSCC, 2021.
- [9] H FUJIWARA. A 5-nm 254-tops/w 221-tops/mm² fully-digital computing-in-memory macro supporting wide-range dynamic-voltage-frequency scaling and simultaneous mac and write operations[C]//ISSCC, 2022.
- [10] ANTONIADIS D D, FENG P, MIFSUD A, et al. Open-source memory compiler for automatic RRAM generation and verification[C]//2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS). IEEE, 2021:97-100.
- [11] ANTONIADIS D, MIFSUD A, FENG P, et al. An Open-Source RRAM Compiler[C]//2022 20th IEEE Interregional NEWCAS Conference (NEWCAS). IEEE, 2022:465-469.
- [12] LIN Z, ZHANG J, WU X, et al. Memory Compiler for RRAM In-Memory Computation[C]//2022 7th International Conference on Integrated Circuits and Microsystems (ICICM). IEEE, 2022:382-385.
- [13] LEE T Y, LEE J M, KIM M K, et al. World-most energy-efficient MRAM technology for non-volatile RAM applications [C]//2022 International Electron Devices Meeting (IEDM). IEEE, 2022:10.7.1-10.7.4.
- [14] JAIN P, KUMAR A, VAN WINKELHOFF N, et al. High Density STT-MRAM compiler design, validation and characterization methodology in 28nm FDSOI technology [C]//2020 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2020:1327-1330.

(责任编辑:薛士然 收稿日期:2023-11-15)