存算一体技术研究现状

李嘉宁,姚 鹏,揭 路,唐建石,伍 冬,高 滨,钱 鹤,吴华强*
(清华大学集成电路学院,北京 100084)

摘 要: 冯诺依曼计算机体系结构面临着“存储墙”的瓶颈,阻碍AI(Artificial Intelligence)计算性能提升. 存算一体硬件结构打破了“存储墙”的限制,大大提升了AI计算的性能. 目前存算一体计算方案已在多种存储介质上得到实现,根据计算信号类型,可以将存算一体计算方案分成数字存算一体方案和模拟存算一体方案. 存算一体硬件结构使得AI计算的性能取得巨大提升,然而进一步发展仍面临重大挑战. 本文对不同信号域的存算一体方案的进行了对比分析,指出了每一种方案的主要优缺点,也指明了存算一体技术面临的挑战. 我们认为,随着工艺集成、器件、电路、架构,软件工具链的跨层次协同研究发展,存算一体技术将在边缘端和云端,为AI计算提供更加强大和高效的算力.

关键词: 人工智能;存算一体;存储介质;计算信号类型;评价指标
基金项目: 国家自然科学基金(No.92164302,No.62025111)
中图分类号: TP389.1 **文献标识码:** A **文章编号:** 0372-2112(2024)04-1103-15
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20230967

Research Status of Computing-in-Memory Technology

LI Jia-ning, YAO Peng, JIE Lu, TANG Jian-shi, WU Dong, GAO Bin, QIAN He, WU Hua-qiang*
(*School of Integated Circuits*, *Tsinghua University*, *Beijing* 100084, *China*)

Abstract: Von Neumann computer architecture faces the bottleneck of “storage wall”, which hindering the performance improvement of AI (Artificial Intelligence) computing. Computing-In-Memory (CIM) breaks the limitation of “storage wall” and greatly improves the performance of AI computing. At present, CIM schemes have been implemented in a variety of storage media. According to the type of calculation signal, CIM scheme can be divided into digital CIM and analog CIM scheme. CIM has greatly improved the performance of AI computing, but the further development still faces major challenges. This article provides a detailed comparative analysis of CIM schemes in different signal domains, pointing out the main advantages and disadvantages of each scheme, and also pointing out the challenges faced by CIM. We believe that with the cross level collaborative research and development of process integration, devices, circuits, architecture, and software toolchains, CIM will provide more powerful and efficient computing power for AI computing at the edge and cloud ends.

Key words: artificial intelligence; computing-in-memory; storage media; calculate signal type; evaluation index
Foundation Item(s): National Natural Science Foundation of China (No.92164302, No.62025111)

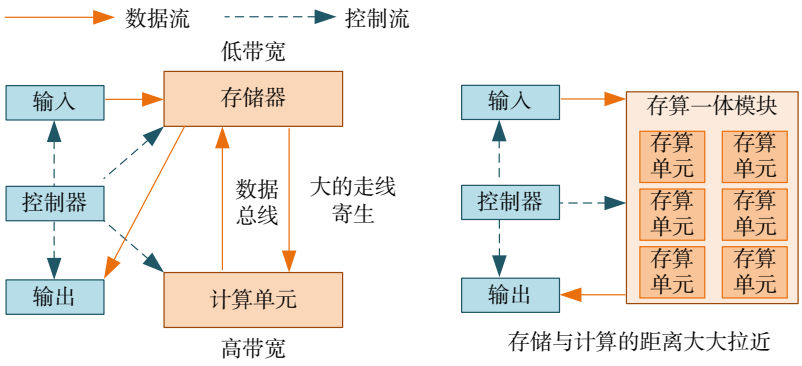
1 引言

现如今,AI(Artificial Intelligence)已经渗透到人类生活的多个方面,成为人类社会不可或缺的一部分. 一大批人工智能技术包括图像分类、物体识别、自然语言处理,生物结构预测等已被开发并被广泛应用[1-4]. 人工智能技术的关键在于神经网络计算的实现. 目前神经网络计算根据计算规模以及部署的位置可分成云计算和边缘计算. 云端计算往往由巨大的数据中心实现,需要大量的计算单元实现超高的算力[5]. 云计算存在数据交互延时大以及个人数据隐私性差的缺陷,很多云端的计算已经向边缘测转移[6]. 边缘计算的应用场景日益复杂,很多边缘计算场景也需要具有可观的算力. 另外考虑到边缘计算设备大多采用电池供电,要求边缘计算也需要有高的能效.

目前主流的计算机体系结构是冯诺依曼体系结构. 如图1(*a*)所示,该结构的存储单元和计算单元是分离的. 在进行数据处理时,存储单元与计算单元要通过总线进行频繁地通信. 具体来说便是数据从存储单元读出通过总线送到计算单元进行计算,计算结果再通

收稿日期:2023-10-19;修回日期:2024-04-03;责任编辑:宋甲英
*通讯作者:吴华强

过总线传送到存储单元实现存储. 存储单元和计算单元的带宽存在严重的不匹配,“存储墙”大大限制了冯诺依曼体系结构计算吞吐[7]. 除此以外,数据在存储单元和计算单元之间的频繁搬运消耗大量能量,从而使计算能效的提升受到限制. 传统的冯诺依曼体系结构不利于云计算和边缘计算带宽和能效的进一步提升.



(*a*) 冯诺依曼体系典型结构 (*b*) 存算一体计算结构

图1 冯诺依曼体系结构与存算一体计算结构对比

为了克服传统冯诺依曼体系结构由于存算分离在计算吞吐和能效上面临的瓶颈,近年来存算一体的硬件结构得到了持续的发展. 存算一体的硬件结构打破了传统冯诺依曼计算结构的限制[8,9]. 如图1(*b*)所示,在该计算结构中,存储单元和逻辑单元的距离被大大拉近或者完全消除,大大减少了数据在存储单元和计算单元之间的传输,显著地降低了计算延时和能耗.

近年来AI的发展使得AI计算性能受到了人们的广泛关注. 存算一体的计算结构十分有利于图2所示的以输入和权重值做乘加计算为主的神经网络计算性能的提升,成为大幅度提升AI计算性能的重要方法. 至今已有多种存算一体方案被提了出来,使得AI计算任务的性能取得了显著的提升. 但同时,存算一体架构的进一步发展仍面临器件、电路和系统层面的严峻挑战.

在下文中,我们将对不同信号域的存算一体方案进行对比分析,并指出每一种方案的主要优缺点,同时也会指明存算一体技术面临的挑战. 我们认为,随着工艺集成、器件、电路、架构,软件工具链的跨层次协同研究发展,存算一体技术将在边缘端和云端,为AI计算提供更加强大和高效的算力.

2 不同存储介质下的存算一体实现方案

近年来,基于不同的存储介质,产生了多种存算一体计算方案. 根据存储介质存储的信息在存储单元掉电以后是否丢失,可将存储器分为易失存储器和非易失存储器.

2.1 易失存储器存算一体方案

2.1.1 SRAM存算一体方案

如图3所示,SRAM(Static Random Access Memory)

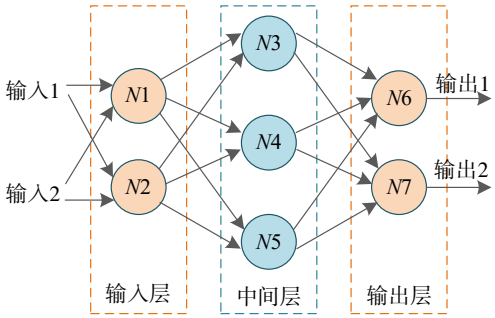


图2 典型神经网络示意图

单元的典型结构由六个金属氧化物场效应晶体管(Metal Oxide Semiconductor,MOS)组成. 四个MOS管组成两个反相器,一个反相器的输出节点同时作为另一个反相器的输入节点,因此形成正反馈环路将数据以电平高低的形式存储在一个反相器的输出节点上. 其余的两个MOS管作为选择器,在读写的过程中选中SRAM单元. 由于数据值以电平高低的形式存储在电路节点上,一旦电路掉电,存储单元存储的数据也会随之丢失.

一个SRAM单元用来存储一个二值的乘数. 为了实现计算,需要给SRAM单元增添相应的计算电路. 电荷共享是SRAM实现计算的一种经典方式. 如图3所示,一种典型的电荷共享型的SRAM计算电路为2T1C结构[10]. SRAM阵列进行计算时,SRAM单元存储的权重值以及输入值在一个2T1C单元上实现相乘,乘法结果以电荷的形式存储于电容上,之后将所有的电容并联,通过电容的电荷共享从而实现各个乘法结果的累加,累加结果为电压的形式,由模数转换电路(Analog to Digital Converter,ADC)进行模数转化. 也有多种其

他结构的SRAM计算电路被提了出来. Si等[11]提出了一种电流域的SRAM计算电路,该计算电路的核心由两个串联的MOS管构成. 计算时,SRAM单元存储的权重值和输入值分别控制计算电路两个MOS管的栅压以控制两个串联的MOS管产生电流的大小从而实现乘法计算,不同行的MOS管电流汇聚同一根BL上实现加法计算. Wu等[12]提出了一种时域的SRAM计算电路,该计算电路的核心由一个两级反相器链组成,SRAM单元存储的权重值以及输入值分别通过控制第一级反相器下拉通路电阻的有无以及阻值大小决定该反相器链是否产生延时从而实现乘法计算,不同反相器链串联从而实现加法计算.

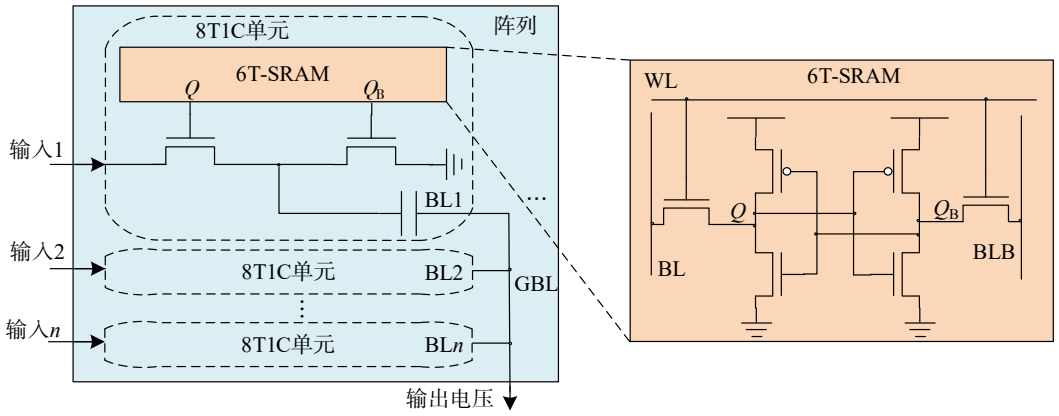


图3　典型SRAM存算一体计算结构以及SRAM单元结构

另外基于SRAM的数字域存算一体技术近年来受到了人们的广泛关注[13~16]. 一种典型的SRAM计算电路为或非(nor)逻辑门[13]. 一个6T结构的SRAM和一个或门组成一个阵列单元. SRAM存储的权重值以及输入值在或门上实现乘法,乘法结果送至阵列外的加法器树实现乘法结果的累加从而实现乘加计算.

SRAM与CMOS工艺完全兼容,和其他的存储器相比,具有工艺成熟度高的特点. 数字域SRAM存算一体方案完全采用逻辑电路进行乘加计算,对噪声不敏感,可以实现高精度的计算[13]. 模拟域的存算一体方案的乘加信号精度和模数转换的精度受工艺扰动影响严重,更适合使用在端侧场景. 作为一种易失性的器件,SRAM在使用的过程中需要持续供电从而形成显著的漏电流,因此SRAM存算一体方案不适用于需求高待机时长的应用场景. 另外,受限于大的SRAM单元面积,SRAM存算一体方案具有较低的存储密度.

2.1.2　eDRAM存算一体方案

一个MOS管和一个电容串联是eDRAM(embeded Dynamic Random Access Memory)单元的一种典型结构[17]. 数据以电荷的形式存储在电容上,MOS管作为选择器用于该单元读写的选中. 电容上的电荷会通过MOS管漏电,因此eDRAM单元需要进行定期的刷新以防止数据的丢失.

eDRAM阵列进行计算时,一个eDRAM单元可实现一个乘法计算. 电荷共享是eDRAM阵列实现计算的经典方式[18]. 权重值以电荷的形式存储在单元的电容上,输入值以电平高低的形式控制单元的MOS管是否导通. 如图4所示,进行乘法计算的单元并联到一根比特线(Bit Line,BL)上,通过共享电容电荷的方式实现乘法结果的累加. 当单元的存储电荷为0或单元的MOS管未导通时,乘法结果为0,在电荷共享的过程中不会提供电荷. 电荷共享所得的电压由ADC进行模数转化.

Zhao等[19]使用2T结构的eDRAM单元实现了乘法计算. 2T结构eDRAM单元的一个MOS用于实现写操作,另一个MOS管用于计算. 在计算模式,eDRAM单元存储的权重值控制计算MOS管是否打开、输入值控制计算MOS管的源漏是否产生压差以控制打开的计算MOS管是否产生电流从而实现乘法计算,同一行(列)上的eDRAM单元的计算电流汇聚在同一根BL上从而实现加法计算. 与上述工作实现乘加计算原理类似,Yu等[20]采用了4T2C的结构实现3值权重的存储与计算.

eDRAM存算一体技术一般具有高的存储密度. 然而,eDRAM存算一体计算具有低的计算吞吐,定期刷新的需求也限制了其在高待机时长场景中的应用.

2.2　非易失存储器存算一体方案

2.2.1　Flash存算一体方案

如图5所示,与MOS器件结构不同,Flash器件在栅级下引入了浮置栅极[21]. 当浮栅上存储电子时,Flash的阈值电压会被抬高,Flash器件难以导通,表示存储数据“0”;当浮栅上未存储电子时,Flash的功能与MOS管类似,在栅极电压为高电平时,管子会导通,表示存储数据“1”. 浮栅被绝缘层环绕,其存储的电子不会轻易流失. 因此即使电路掉电,Flash存储的数据也不会丢失.

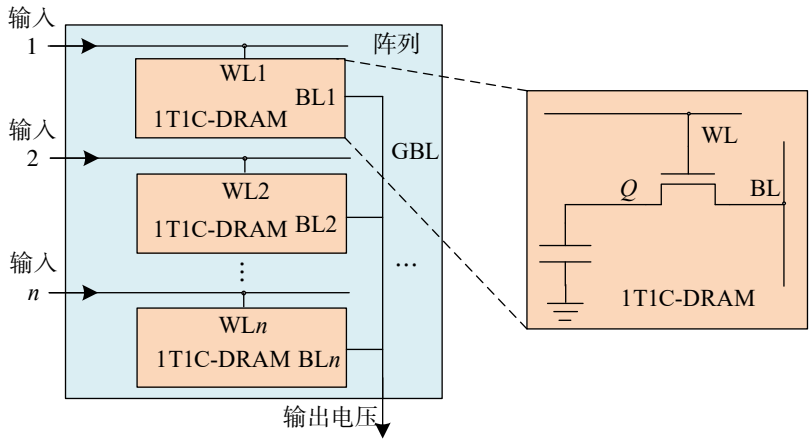


图4 典型的DRAM存算一体计算结构以及DRAM单元结构

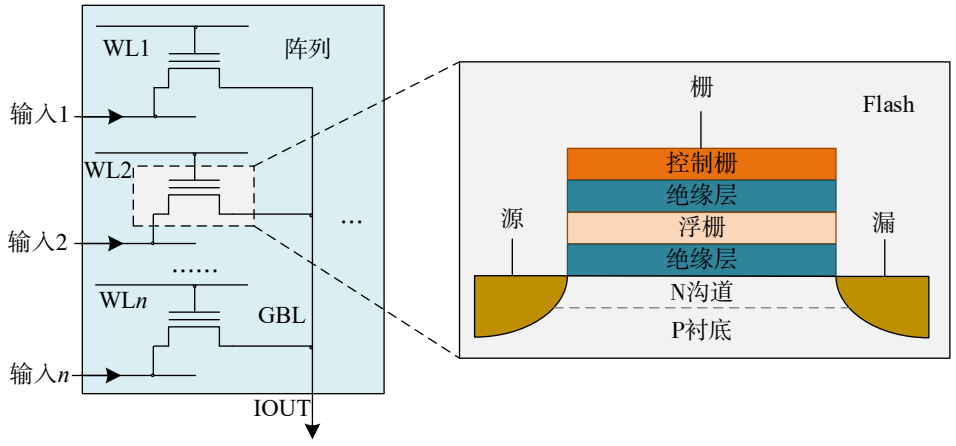


图5 典型的Flash存算一体计算结构以及Flash器件结构

Flash阵列进行计算时，一个Flash单元可以实现一个乘法计算. 如图5所示，在Flash器件上加电压产生乘法电流是Flash单元实现乘法计算的典型方式[22]. 权重值以浮栅上电子的有无存储在Flash器件上. 输入值通过数模转换器(Digital to Analog Converter，DAC)转换成模拟电压施加在Flash的源级，因此会在工作在线性区的Flash器件的漏极形成乘法电流. 多个Flash的乘法电流汇聚在一根SL上从而实现电流累加并经ADC转化成数字信号. 另外，通过分别控制Flash浮栅上电子的数量以及漏极电压的level数量可以在一个Flash单元上实现多bit输入、多bit权重的乘法.

Flash器件作为传统的非易失器件，相比其他的非易失器件工艺成熟度更高，可实现多bit存储，在存储密度上相比其他的非易失器件具有更加明显的优势，可用在高待机时长、低功耗的边侧和端侧场景. 相比于其他非易失器件，平面Flash工艺的进一步微缩面临浮栅中电子数量难以控制的困难；另外Flash器件擦写次数更少，且读取速度更慢.

2.2.2 忆阻器存算一体方案

近年来，忆阻器在存储以及计算领域受到了研究者的广泛关注. 忆阻器通过器件阻值的高低存储信息，根据忆阻器阻变机理的不同，如图6所示，可以将忆阻器划分为阻变随机存储器、相变存储器、以及磁阻变随机存储器等. 阻变随机存储器(Resistive Random Access Memory，RRAM)一般是金属-绝缘体-金属结构，通过控制绝缘体内导电细丝的断裂与形成实现高低阻态的转换从而实现存储信息的转换[23]. 相变存储器(Phase Change Memory，PCM)的结构与阻变随机存储器结构类似，然而在存储状态的转变原理上具有很大的不同[24]. 相变存储器是通过控制与下电极连接处的相变材料的状态在晶体和非晶体之间切换实现器件高低阻态的切换. 磁阻变随机存储器(Magnetoresistive Random Access Memory，MRAM)一般包括自由磁层、隧道栅层、以及固定磁层，通过控制自由磁层的磁场方向与固定磁层的磁场方向是否平行实现器件高低阻态的切换[25]. 忆阻器编程一旦结束，器件的阻值状态会维持很长时间，除非给器件施加编程操作条件. 因此，断电以后，器件存储的信息也不会丢失. 忆阻器往往会与

一个MOS管串联组成1T1R结构作为忆阻器阵列的基本单元结构,MOS主要作为选择器控制读写过程忆阻器的选中.

1T1R忆阻器阵列进行计算时,一个单元可以实现一个乘法计算. 如图6所示在1T1R单元上施加电压产生乘法电流是1T1R单元实现乘法计算的典型方式. 权重值以忆阻器电导值的形式存储于忆阻器器件. 输入值通过DAC转换成模拟电压施加在1T1R单元的BL端,该电压与源线(Sourse Line,SL)端的压差与忆阻器电导相乘得到一个乘法电流流入SL,多个1T1R单元的电流汇聚在同一根SL上从而实现乘法电流的累积,乘加电流由ADC转化成数字信号[26].

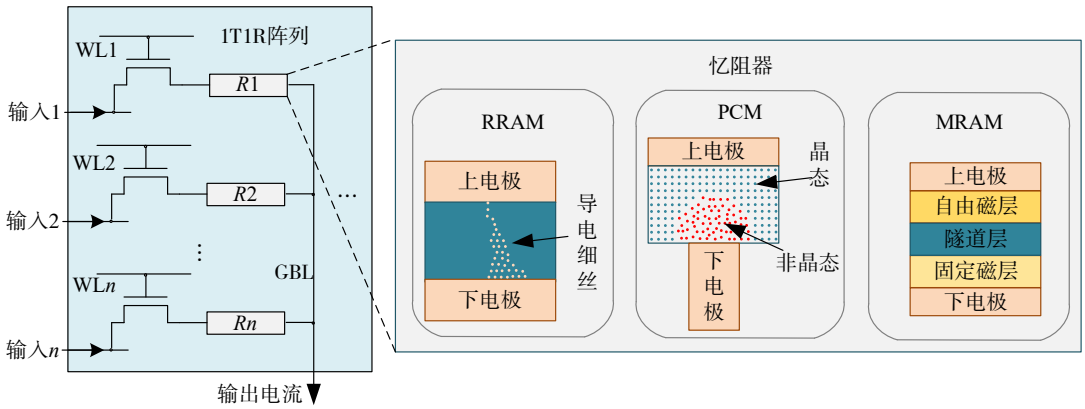


图6　典型的忆阻器存算一体计算结构以及不同种类的忆阻器器件结构

相比RRAM和PCM,MRAM在耐久性、读取速度上具有明显的优势;然而MRAM的高低阻态的阻值之比要小于RRAM和PCM,尚未有MRAM进行多bit存储的报道,而已有报道在RRAM[27]和PCM[28]上可实现多bit的权重存储. 另外与PCM器件相比,RRAM低成本、与CMOS工艺更加兼容的特点使得其受到了广泛关注. 基于RRAM、PCM、MRAM器件的存算一体方案在端测和边缘计算场景具有很大的潜力,然而大规模应用的实现还依赖于器件工艺水平进一步提升. 上述各种存储器件特性对比如表1所示.

表1　存储器件特性对比

存储器类型	SRAM[15]	eDRAM[29]	FLASH[30]	RRAM[31]	PCM[24]	MRAM[32]
存储密度	低	很高	很高	高	高	高
读取速度	很快	快	中	中	中	快
易失性	易失	易失	非易失	非易失	非易失	非易失
开关比	很高	很高	高	高	高	低
耐久性	很高	很高	低	中	中	高
多bit存储	否	否	是	是	是	否

3　不同信号类型的存算一体方案

根据计算信号的类型,可以将存算一体方案划分成数字域的存算一体方案以及模拟域的存算一体方案. 根据模拟信号的种类可将模拟域的存算一体方案进一步划分为电流域、电荷域、时域的存算一体方案.

3.1　数字域的存算一体方案

数字域的存算一体方案常常采用逻辑电路实现乘法和加法. 得益于逻辑电路计算功能的鲁棒性,数字域的存算一体方案可以取得更高精度的计算结果[13,14];使用逻辑电路实现乘加计算也使得数字域存算一体方案往往具有快的单次计算速度. 然而,数字域的存算一体方案也面临着一些挑战. 全逻辑电路实现的外围乘加电路具有大的计算面积,在实际的数字域存算一体方案中往往采用乘加电路复用的方式以减小外围电路的面积占比,却付出了计算吞吐下降的代价;数字域存算一体方案中的一个存储单元往往只存储1 bit数据,数据存储密度相对较小;一次操作实现多bit数据的乘加计算需要付出大的外围电路面积,于是数据输入常常采用单bit输入的方式,却降低了数据吞吐. 有多种优化方案被提出以克服上述所提的挑战.

表2列举了一些典型的数字域存算一体方案. 台积电提出了一种数字域的SRAM存算一体计算方案[13]. 该方案中6T结构的SRAM紧邻一个nor门构成了10T的阵列单元结构. 在进行计算时,单元内SRAM存储节点的电平作为nor门的一个输入,输入值以高低电平的形式作为nor门的另外一个输入,实现乘法计算. 每一组乘法结果在阵列外的加法器树上进行累加,得到最终的乘加计算结果. 为了减小加法器树的开销,加法器树中的全加器采用了28T结构和14T结构间隔的方式排布,从而节省了部分面积和能量的开销. 不过即使如此,加法器树的开销仍然很大.

新竹清华大学提出了一款数字域的MRAM存算一体方案[33]. 进行计算时,输入值由外部电路送给阵列外的乘法逻辑单元,权重值从MRAM阵列中经由灵敏放

大器(Sense Amplifier,SA)读出送到阵列外的乘法逻辑单元,由此完成一组乘法计算. 每一组数字域的乘法结果通过加法器树实现累加. 为了加快从阵列中读取数据的速度,该工作提出了一种双向BL读取方案以减小连续读取阵列的读取时间. 同时也设计了一款可将电荷再利用的SA以减小连续读过程中的能量消耗. 由于每次只能读取一行数据,而一个8 bit的权重值存储于8行阵列中,需要读取8次才能得到8 bit的权重值,因此会造成较慢的读取速度.

东南大学提出了一款数字域的SRAM存算一体方案[14]. 进行计算时,将阵列中选中的SRAM的存储的权重值读出到乘法逻辑电路的输入端,同时也将输入数据输入到乘法逻辑电路完成一组乘法计算,然后每一组乘法计算的逻辑结果通过加法器树累加得到乘加结果. 该方案可以支持8 bit的整形计算. 加法器树占据了数字域SRAM存算一体方案的主要面积开销和功耗开销,这篇文章对低6 bit权重数据进行近似乘法计算,从而减小了加法器树的面积和功耗开销. 然而,近似乘法计算的方式对加法器树开销的优化效果是有限的,同时近似计算也会降低数字域计算在计算精度上的优势. 值得一提的是,这项工作同时兼容了浮点型数据的计算以提升神经网络的推理精度.

对浮点数的支持是最近三年存算一体技术向更高精度计算场景拓展过程中发展出的新特征. 不同应用场景对数据的精度需求有所不同,浮点计算仍是以面向训练应用为主,整型计算以面向推理应用为主. 最新发表的一些面向推理的存算一体芯片的工作中,浮点计算的实现往往需要先通过对齐电路将浮点数转化为整数再送至存储阵列,存储阵列上进行的仍然是数字域或模拟域的整形计算[14,16],目前存算一体技术中浮点计算实现的原理和整形计算并没有本质上的区别. 在数据对齐转化为整数的过程中,常常需要截断计算数据的低bit位,因此输入数据的精度有所损失;对于未被截掉的输入数据的部分,在进行计算时,也常常会对低bit位做近似计算[14]或进行模拟域的计算[16]以减小计算开销,因此计算结果的精度也会有所下降. 目前有工作采用精度可变[34]或兼容整形计算[35]的浮点计算架构以应对不同使用场景的需求. 也有工作提供的数据表明,对浮点计算的支持使得逻辑电路变得更加复杂,对计算速度和能效产生负面影响,和整形计算相比,两项指标有所降低[14].

表2 一些典型的数字域存算一体方案

文献来源	文献[13]	文献[33]	文献[14]
工艺/nm	22	22	28
信号类型	数字域	数字域	数字域
容量/Mb	0.064	4	0.064
主要特点	优化加法器树降低开销	BL双向读取降低连续读功耗	引入近似计算减小加法器树规模

3.2 模拟域的存算一体方案

根据模拟信号的种类,可将模拟域存算一体实现方案进一步划分为电流域、电荷域、时域的存算一体方案.

3.2.1 电流域的存算一体方案

电流域的存算一体计算方案一般以电压作为输入信号,电压经由MOS管或电阻器件的跨导转化为乘法电流从而实现乘法计算,乘法电流基于基尔霍夫电流定律在同一根BL上进行累加从而实现加法计算,乘加电流经由ADC转化为数字信号. 电流域的存算一体计算方案十分有利于大规模乘加计算的实现,然而其进一步发展也面临着一些挑战. 大的阵列开启规模会造成大的乘加电流,乘加电流流经BL会在BL上形成电压降,会影响施加在阵列单元上的实际电压,从而降低乘加电流的精度;乘加电流往往需要经由电流-电压转换电路转化为电压信号,信号裕度会有所降低;乘加电流的精度受工艺波动影响严重;高精度的模数转换电路会使得电路开销急剧增加.

表3列举了一些典型的电流域存算一体方案. 清华大学吴华强课题组提出了一款电流域的RRAM存算一体计算方案[36]. 该方案采用了新型的2T2R的阵列单元结构,其中一个RRAM单元存储正权重值,另外一个RRAM单元存储负权重值,因此在一个2T2R单元上实现了带符号的三值权重存储. 每个2T2R单元的正负权重电流在单元内部实现抵消,从而大大降低了多个单元并行计算时在SL上的汇聚电流,减小了SL上的电压降,提升了阵列产生的乘加电流的精度以及阵列并行计算的并行度. 汇聚在SL上的乘加电流由ADC进行模数转化. 这项工作设计了一种转换精度可调的ADC,在不同的应用需求下可以选择不同的ADC的转换精度达到控制计算延时和计算能耗的目的. 在另外的一项工作中,其基于8个2 Kb存储规模的1T1R计算芯片首次搭建了全硬件实现的卷积神经网络[27]. 每一个RRAM单元在32个电导level中进行编程,因此单个1T1R单元实现了5 bit的权重存储. 阵列中1T1R单元的乘法电流在SL汇聚,最终乘加电流由ADC进行模数转化得到数字信号. 这项工作搭建的神经网络包括两层卷积层,两层池化层以及一层全连接层. 该工作采用了混合训练方案,将卷积层和全连接层的权重进行片外训练之后往忆阻器阵列上进行映射,在此之后只对全连接层的权重进行片上训练从而以小的训练代价实现神经网络推理精度的提升.

表3　一些典型的电流域存算一体方案

文献来源	文献[36]	文献[27]
工艺/nm	130	130
信号类型	模拟电流域	模拟电流域
容量/Mb	0.158 8	0.016
主要特点	2T2R单元降低电压降;可调精度的ADC	全硬件实现的卷积神经网络;混合训练减小训练代价
文献来源	文献[28]	文献[11]
工艺/nm	40	28
信号类型	模拟电流域	模拟电流域
容量/Mb	2	0.064
主要特点	权重低位数据采用多值PCM存储均衡精度与存储密度	增加计算电路管子尺寸减小计算信号受工艺波动影响

新竹清华大学提出了一款电流域的PCM存算一体计算方案[28]. 阵列的一个单元为1T1R结构. 权重值以PCM电导值的形式存储于阵列单元上. 进行计算时,输入值会被DAC转换成单元SL与BL电压的差值从而在SL上产生乘法电流实现乘法. 多个1T1R单元的SL共用因此产生的乘法电流汇聚在SL上实现乘法电流的累加. SL的累加电流在阵列外的电阻上形成电压差由ADC进行模数转化得到数字信号. 这篇文章将8 bit的权重乘数用5个阵列单元存储,较好地折衷了存储密度与计算精度. 同时这项工作中的ADC采用了电压捕获型的SA以增加信号检测裕度. 该机构也在SRAM存储介质上实现了电流域的存算一体计算方案[11]. 其在SRAM子阵列外放置了计算单元. 进行计算时,SRAM单元存储的权重值以电平高低的形式施加在计算单元上控制计算单元是否产生电流,输入值会被DAC转换成模拟电压值施加在计算单元控制计算单元产生电流的高低. 多个计算单元上的电流汇聚产生乘加电流,该电流在阵列外的电阻上形成电压差由ADC进行模数转换得到数字信号. 通过多个SRAM复用一个计算单元的形式,计算单元管子面积可增加以减小乘法电流受到工艺扰动的影响从而增加计算精度.

3.2.2　电荷域的存算一体方案

电荷域的存算一体方案常常采用计算电容之间进行电荷共享得到模拟电压的方式实现模拟乘加计算. 金属-氧化物-金属电容具有好的工艺稳定性,且MOS管在此类方案中常作开关使用,因此电荷域的存算一体方案的乘加计算结果受工艺波动影响较小. 电荷域的存算一体方案的计算结果电压的范围可在电源和地之间,具有较大的信号裕度. 然而,电荷域的存算一体方案也面临着高精度模数转换电路开销大的挑战;另外乘加计算的实现过程中电容的使用会造成大的阵列单元面积,使得存储密度显著下降.

表4列举了一些典型的电荷域存算一体方案. 联发科提出了一款电荷域的SRAM存算一体计算方案[37]. 该方案中6T结构的SRAM紧邻3个开关管组成9T的阵列单元. 为了减小阵列单元的面积,计算电容在阵列单元之外被阵列的一行单元共用,该电容在同一时间只会被一个阵列单元使用. 进行计算时,被选中的SRAM中的开关管会根据SRAM存储节点的电平值的高低决定往计算电容传送由输入值经过DAC转换成的电压还是零电位. 计算电容并联进行电荷共享从而实现乘加计算. 电荷共享后在所有的乘加电容的共同底板上得到的电压值由ADC进行模数转换得到数字信号. 由于一个计算电容被多个SRAM复用,因此降低了阵列计算的并行度. 北京大学黄如课题组提出了一款电荷域的SRAM存算一体计算方案[38]. 每一个6T的SRAM单元紧邻一个3T1C的计算电路形成了9T1C的阵列单元结构. 该方案在每一个阵列单元内都嵌入了乘加电容,因此可以在阵列内部实现计算. 这项工作也提出了一种可实现低位优先计算的ADC来降低ADC的转换功耗. 在阵列中的每一个单元上均插入乘加电容虽使得阵列中的每一个单元均可同时参加计算,却造成阵列单元面积太大从而大大降低了存储密度.

3.2.3　时域的存算一体方案

表5列举了一些典型的时域存算一体方案. 新竹清华大学提出了一种时域的SRAM存算一体方案[12]. 该方案以时延产生电路作为乘法电路. 进行计算时,SRAM将存储节点的电平施加于时延电路,另外一个乘

表4　一些典型的电荷域存算一体方案

文献来源	文献[37]	文献[38]
工艺/nm	12	22
信号类型	模拟电荷域	模拟电荷域
容量/Mb	0.128	0.128
主要特点	计算电容阵列共用减小阵列面积	计算电容嵌入阵列单元增加计算带宽

数通过DAC转化成某一level电压施加在时延电路上，一组输入和权重的乘法结果体现为时延电路输入上升沿到输出上升沿的时间延时．通过将时延电路串联，每一个时延电路上的时间延时可自动累加，由此实现了加法计算．时延电路串的总延时由时间数字转换器(Time to Digital Converter，TDC)进行模数转化．该机构在同一年也在RRAM存储介质上实现了时域的存算一体计算方案[39]．阵列的一个单元为1T1R结构．权重值以电导的形式存储在阵列单元中，输入值会转化成电平的高低控制单元的开关管是否打开．阵列进行计算时，会首先将参与计算的1T1R单元共用的SL充电到一个固定电压，然后SL通过阵列单元放电．将某一固定的时间与SL从某一个高电压放电到另一低电压所用的时间相减所得的时间值就是阵列的计算结果．通过TDC电路将该时间转化成数字信号．这项工作将阵列上的直流静态功耗转换成了动态功耗，从而减小了阵列能量消耗．时域的计算信号相比其他模式的模拟计算信号有更大的信号裕度，时域的模数转换电路也有更小的面积以及更低的功耗．然而，时域乘加信号的准确度会严重受到工艺波动影响．

表5 一些典型的时域存算一体方案

文献来源	文献[12]	文献[39]
工艺/nm	28	22
信号类型	模拟时间域	模拟时间域
容量/Mb	1	8
主要特点	小的时域模数转换电路开销;大的时域信号margin	小的阵列功耗开销,大的时域信号margin

4 计算性能的评价指标

4.1 算力

算力指标用来评价系统计算的速度，单位为OPS．其物理意义表示系统在单位时间内可做计算的次数．对于计算系统来说，计算次数表示乘法计算次数与加法计算次数之和．计算算力可用式(1)计算：

$$\text{算力}=\frac{2\times\text{输入并行度}\times\text{输出并行度}}{\text{一次计算时间}} \tag{1}$$

我们将近年的一些典型存算一体方案取得的算力归一化为1 bit输入、1 bit权重计算下的等效算力．如图7所示，近年来存算一体方案取得的算力逐年快速提升．

4.2 能量效率

能量效率的单位为OPS/W．其物理意义表示系统单位能量可做计算的次数．能量效率可以用式(2)进行计算：

$$\text{能效}=\frac{2\times\text{输入并行度}\times\text{输出并行度}}{\text{一次计算能量}} \tag{2}$$

同样地，我们将各个存算一体方案的能效指标归一化为1 bit输入、1 bit权重计算下的等效能效．如图8所示，近年来存算一体方案取得的能效呈现逐年提升的趋势．

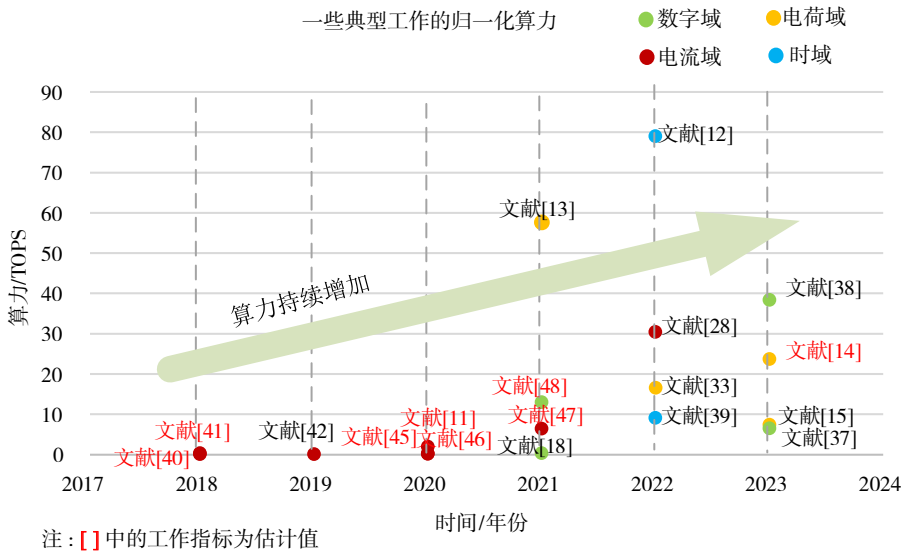


图7 存算一体算力发展趋势

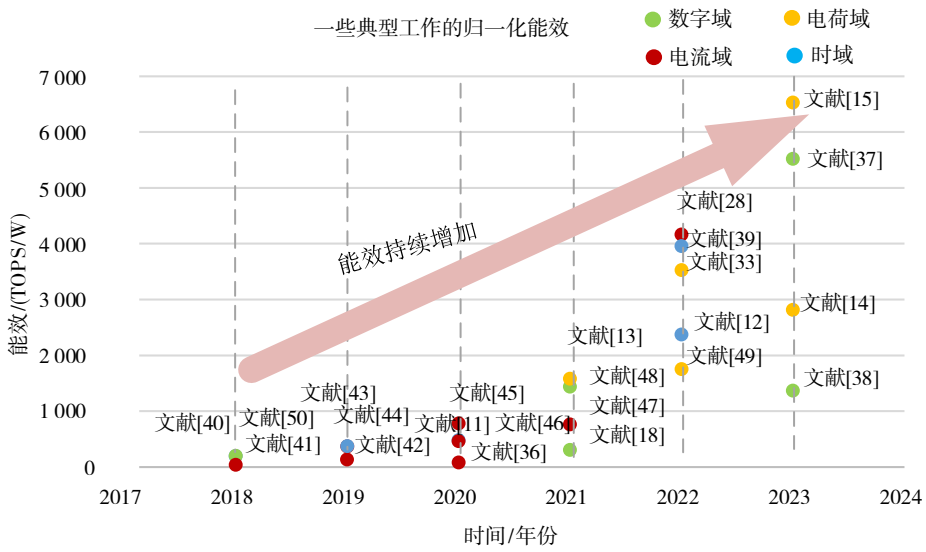


图 8　存算一体能效发展趋势

4.3　面积效率

芯片面积的大小决定了芯片的成本,因此计算方案的设计也需要考虑面积效率.

4.3.1　算力密度

我们可以想到一个场景,如果只为增加算力,我们可以通过将原先的芯片复制一份使两个芯片同时工作的方式实现,然而这样会造成芯片面积的加倍从而提高成本. 为了能更全面地评估算力指标,算力密度指标被提了出来. 算力密度指标表示单位芯片面积可以达到的算力,单位为 OPS/mm^2. 如式(3)所示,算力密度指标可通过芯片算力除以芯片面积得到:

$$\text{算力密度}=\frac{\text{算力}}{\text{芯片面积}} \tag{3}$$

4.3.2　存储密度

更加复杂的神经网络往往具有更多的权重数量. 为了芯片能够处理更加复杂的神经网络,我们希望芯片可以存储更多的数据. 我们可以通过直接增加存储容量的方式增加芯片存储的数据量,然而这样会直接造成芯片面积增加. 为了能更好地评价芯片的数据存储能力,存储密度的指标被提了出来. 存储密度指标表示单位芯片面积可以存储的数据量,单位为 Mb/mm^2,如式(4)所示,可通过芯片存储的数据量与芯片面积相除计算得到:

$$\text{存储密度}=\frac{\text{存储容量}}{\text{芯片面积}} \tag{4}$$

我们定性比较了不同信号域的各个指标的高低情况,如表 6 所示.

4.4　综合评价指标

在存算一体芯片的设计过程中,一个指标的提升常常要以另外一些指标的下降为代价. 为了能够评价芯片的综合性能. 在最近的一些工作中,有一些更加综合的评价指标被提了出来. 北京大学黄如教授等提出可以用能效和存储密度的乘积来对芯片进行综合评价[39]. 在另一项工作中,黄如等提出可以用能效和算力密度的乘积对芯片的综合性能进行评价[37]. 张孟凡等提出了一些将输入输出数据精度考虑在内的评价指标,以强调计算数据的精度在计算综合性能评估上的重要性[51]. 不同的应用场景对芯片有不同的性能需求,性能的优化方向也会有所不同,从而产生了不同的评价指标.

表 6　不同信号域的各个指标对比

计算信号类型	数字域	电流域	电荷域	时域
算力密度	中	高	中低	中
能量效率	高	中	高	高
存储密度	中	高	中低	中
信号裕度	高	中低	中	中高

注:具体方案的指标可能根据优化方式的不同有所区别

5　存算一体技术面临的主要挑战

5.1　读扰乱问题

多行同时开启增加阵列计算并行度以提高计算带宽的方式会造成 SRAM 单元的值在计算过程中被改写的问题. 例如,多行 SRAM 同时打开时,如果存储数据“0”的 SRAM 过多,会使得共用 BL 电压下降到 SRAM 单元写数据“0”的阈值电压内,因此造成存储数据“1”的 SRAM 的存储值被改写. 从而造成读扰乱问题. 读扰乱问题大大降低了 SRAM 阵列可同时进行计算的并行度从而降低计算带宽. 为了解决读扰乱的问题,通常控制计算过程 BL 电压位于写“0”阈值电压以上[50,52],另外也有一些非 6T 结构的 SRAM 单元被提了出来[53~55]. 非

易失器件的存储状态会受到外部施加电压的影响，外部施加的电压越高，非易失器件的存储状态就越容易发生改变[56~59]. 在计算过程中，非易失器件加压过大就会造成读扰乱的问题，因此要将读电压限制在一个较小的范围内. 这样往往造成输出信号裕度的下降，并对输入电压的准确度要求更高而增加外围电路的开销.

5.2 权重单元密度有限

如前所述，SRAM存算一体方案为了防止读扰乱问题常常使用了非6T结构的SRAM单元结构，如8T[53]、9T[37]、10T[55]、9T1C[38]结构等. 这些结构在解决读扰乱问题并赋予SRAM单元计算功能的同时，也大大增加了SRAM阵列单元的面积，降低了芯片的存储密度. 对大多数非易失存储器件的存算一体方案来说，为了减小阵列的电压降问题对计算信号精度的影响，常常需要更粗的阵列走线，这会使得存算一体计算芯片中的存储单元的面积大于存储器芯片中的阵列单元的尺寸[36]. 值得一提的是，采用多值存储而非权重的空间展开，能进一步提升权重密度[27,28].

5.3 计算电路开销大

在数字域计算中，加法器树占据了计算电路的主要面积，加法器电路是数字域计算方案的优化重点[15]. 一些工作通过简化加法器的结构以减小加法器的开销[13]，也有一些工作通过近似计算减小加法器的处理的数据宽度以减小加法器树的开销[14]，这样的方式取得的优化效果是有限的. 另外有一些工作通过减少加法器树个数以减小芯片面积，然而这样就造成了计算带宽的大大降低. 在模拟域计算中，高精度的模数转换电路成为了计算性能提升的瓶颈[60,61]. 逐次逼近寄存器型ADC相比于并联型ADC更适用于高精度下的模数转换场景[62]. 然而在高精度模数转换场景下，逐次逼近寄存器型ADC也面临着转换次数多导致的计算周期长，精度要求高导致的电路面积增加的问题. 一些工作通过增加ADC的复用度的方式以减小ADC的总开销，却以大大牺牲计算带宽为代价[63]. 另一些工作通过使用低精度的ADC以降低ADC的开销，却付出了计算精度下降的代价[64]. 表7中列出了一些典型的存算一体方案中外围计算电路的面积和功耗占比.

表7 一些工作计算电路开销占比

文献来源	文献[15]	文献[60]	文献[38]
乘加信号类型	数字域	电流域	电荷域
计算电路面积占比	53%	40.26%	30%
计算电路功耗占比	38.4%	50.36%	

5.4 多bit计算实现的开销大

很多应用场景需要神经网络对复杂数据集有高的推理精度，这就需要输入数据和权重数据具有更高精度. 对于SRAM存算一体方案来说，一个多bit权重数据需要多个SRAM单元进行存储[11,13,38]，因此减小了阵列可以存储的权重个数. 对于非易失器件的存算一体方案来说，目前已经有多种非易失器件的存储状态可以在多个level之间进行编程从而一个阵列单元便能实现多bit数据存储[27,28]，然而这种方式会降低阵列乘加信号的精度. 对数字域的存算一体方案来说，多bit的输入数据往往需要多个计算cycle以完成一次完整的多bit输入的计算[38]，从而大大增加了计算的时间；要实现多bit数据的输入，通常需要将计算电路的数量加倍因此造成计算电路面积的显著增加[14]. 对模拟域的存算一体方案来说，多bit数据通常需要DAC进行数模转化得到模拟电压施加在阵列上[11,37]，为了保证乘加信号精度，往往以增加外围电路的面积和功耗为代价.

5.5 计算精度受工艺波动影响大

数字域的存算一体技术方案借助逻辑电路对工艺波动的鲁棒性，可以实现接近软件的计算精度[16]. 在基于SRAM的模拟存算一体计算方案中，电流域、时域的乘法计算的实现往往利用MOS管的电流源特性或电阻特性. 工艺波动会造成MOS管的阈值电压的波动，在同一栅极电压下，不同的MOS管产生的电流值[61]或体现出的电阻值[12]也会随之波动，于是其对应的实际权重值和映射的理想权重值之间存在差异，从而在工艺波动的影响下，网络推理精度有所下降. 已有工作为了减小工艺扰动对模拟计算结果的影响，采用增加了阵列中计算电路的面积的方法，却减少了计算电路的数量[61]. 除了对计算单元的计算结果造成影响以外，工艺波动会造成外围数模转换和模数转换电路的转换精度进而影响网络精度. 已有工作致力于提升外围转换电路的精度，却仍有近1%的错误率[65].

在基于非易失器件的模拟存算一体方案中，除了PVT扰动导致的外部电路精度下降外，还会引入器件的非理想因素. 工艺的缺陷以及随机性使得不同非易失器件的性能有所波动，在面向训练的存算一体芯片中不同器件之间的编程特性不一致，使得训练精度提升面临着大的挑战[62,66]；在面向推理的存算一体芯片中，器件常常表现出阻值漂移特性，一方面使得单器件对应的实际权重值逐渐偏离理想权重值，另一方面不同器件的阻值漂移特性也难以保持一致. 一项工作直观展示了器件电导值的平均值随着时间推进不断偏离理想电导值，且偏离的方差值随时间推进逐渐增加，网络精度也随时间逐渐下降[67]. 有工作尝试通过增加阵列单元产生参考电流以做漂移补偿，然而不同器件漂移特性的随机性使得补偿效果有限[68].

5.6 多核芯片高效流水实现困难

除了对存算一体宏电路模块的性能进行优化以

外,也需考虑芯片系统级的性能优化. 如表8所示[69],多核的系统级芯片(System On Chip,SOC)的整体性能,除了受到单核性能的影响以外,也受到芯片内部各个部分之间通信性能的影响. 在芯片架构层面,我们需要在芯片的各个部分之间实现高能效,低延时,高带宽的数据传输以获得好的芯片级的性能. 片上互联方式的片上通信网络与总线方式的片上通信网络相比在可扩展性上有更大的优势[70]. 然而大规模的片上互联会增加死锁以及活锁的出现概率,同时在面积和能耗开销上相比总线方式更大. 总线和片上网络的通信方式均还不能太好地满足多核芯片的性能需求. 目前已有一些工作尝试对片上互联通信网络的功耗进行优化[71],也有一些工作尝试增强片上互联网络的容错性[72].

表8　不同系统层级的能效对比[69]

系统层级	单tile	单chip	全系统
能效/(TOPS/W)	20.0	12.4	6.94

5.7　软件工具链还需进一步优化

目前已经有了多种仿真工具可分别在不同的设计层次上对存算一体芯片的性能进行评估. CACTI和Orion工具可用来评估RRAM存算芯片外围电路的功耗和面积开销[73];MNSIM,NeuroSim[74],XB-sim[75]可用于评估阵列级的功耗,面积和延时. 目前除了单个仿真工具的仿真性能需要进一步提升以外,实现已有的各个仿真器之间的兼容也是十分重要的. 除了仿真器外,标准化的编译器也需要进一步优化和改进.

6　总结与展望

存算一体打破了传统计算结构“存储墙”的限制,大大提升了AI计算的性能. 存算一体方案已经在多种存储介质下的得到实现,根据计算信号的类型将存算一体计算方案分别划分为数字域,电流域、电压域、电荷域、时域的存算一体计算方案. 存算一体技术在多个层面上面临着很多挑战. 总的来说,在器件层面,非易失器件稳定的多值存储能力具有很强的吸引力,还需要在材料和器件结构层面做更深入的研究;在电路层面,计算数据的高bit位进行数字计算、低bit位进行模拟计算的混合计算方式可能成为均衡计算精度和电路开销更好的方式;在架构层面,多种互联形式并存成为芯片内各个模块通信的发展趋势;在软件层面,集成的、通用的软件工具链的实现对提升存算一体芯片设计效率和部署效率方面具有十分重要的意义. 值得一提的是,近年来芯粒(chiplet)技术也为存算一体系统性能的提升提供了重要的思路[76]. 随着工艺集成、器件、电路、架构,软件工具链的跨层次协同研究发展,存算一体技术将在边缘端和云端,为AI计算提供更加强大和高效的算力.

参考文献

[1] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.

[2] SENIOR A W, EVANS R, JUMPER J, et al. Improved protein structure prediction using potentials from deep learning[J]. Nature, 2020, 577(7792): 706-710.

[3] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2016: 770-778.

[4] SONG X, ZOU Y X, HUANG S L, et al. Investigating multi-task learning for automatic speech recognition with code-switching between mandarin and English[C]//2017 International Conference on Asian Language Processing (IALP). Piscataway: IEEE, 2017: 27-30.

[5] ASIF-UR-RAHMAN M, AFSANA F, MAHMUD M, et al. Toward a heterogeneous mist, fog, and cloud-based framework for the internet of healthcare things[J]. IEEE Internet of Things Journal, 2019, 6(3): 4049-4062.

[6] ABBAS N, ZHANG Y, TAHERKORDI A, et al. Mobile edge computing: A survey[J]. IEEE Internet of Things Journal, 2018, 5(1): 450-465.

[7] WULF W A, MCKEE S A. Hitting the memory wall[J]. ACM SIGARCH Computer Architecture News, 1995, 23(1): 20-24.

[8] ZHANG D P, JAYASENA N, LYASHEVSKY A, et al. TOP-PIM: Throughput-oriented programmable processing in memory[C]//Proceedings of the 23rd International Symposium on High-performance Parallel and Distributed Computing. New York: ACM, 2014: 85-98.

[9] AHN J, HONG S, YOO S, et al. A scalable processing-in-memory accelerator for parallel graph processing[C]//Proceedings of the 42nd Annual International Symposium on Computer Architecture. New York: ACM, 2015: 105-117.

[10] JIANG Z W, YIN S H, SEO J S, et al. C3SRAM: An In-memory-computing SRAM macro based on robust capacitive coupling computing mechanism[J]. IEEE Journal of Solid-State Circuits, 2020, 55(7): 1888-1897.

[11] SI X, TU Y N, HUANG W H, et al. 15.5 A 28nm 64Kb 6T SRAM computing-in-memory macro with 8b MAC operation for AI edge chips[C]//2020 IEEE International Solid-State Circuits Conference - (ISSCC). Piscataway: IEEE, 2020: 246-248.

[12] WU P C, SU J W, CHUNG Y L, et al. A 28nm 1Mb time-domain computing-in-memory 6T-SRAM macro with a 6.6ns latency, 1241GOPS and 37.01TOPS/W for 8b-MAC operations for edge-AI devices[C]//2022 IEEE International

Solid-State Circuits Conference (ISSCC). Piscataway: IEEE, 2022: 1-3.

[13] CHIH Y D, LEE P H, FUJIWARA H, et al. 16.4 an 89TOPS/W and 16.3TOPS/mm^2 all-digital SRAM-based full-precision compute-in memory macro in 22nm for machine-learning edge applications[C]//2021 IEEE International Solid-State Circuits Conference (ISSCC). Piscataway: IEEE, 2021: 252-254.

[14] GUO A, SI X, CHEN X, et al. A 28nm 64-Kb 31.6-TFLOPS/W digital-domain floating-point-computing-unit and double-bit 6T-SRAM computing-in-memory macro for floating-point CNNs[C]//2023 IEEE International Solid-State Circuits Conference (ISSCC). Piscataway: IEEE, 2023: 128-130.

[15] HE Y F, DIAO H K, TANG C, et al. 7.3 A 28nm 38-to-102-TOPS/W 8b multiply-less approximate digital SRAM compute-In-memory macro for neural-network inference[C]//2023 IEEE International Solid-State Circuits Conference (ISSCC). Piscataway: IEEE, 2023: 130-132.

[16] WU P C, SU J W, HONG L Y, et al. A 22nm 832Kb hybrid-domain floating-point SRAM in-memory-compute macro with 16.2-70.2TFLOPS/W for high-accuracy AI-edge devices [C]//2023 IEEE International Solid-State Circuits Conference (ISSCC). Piscataway: IEEE, 2023: 126-128.

[17] KIM S, KIM S, UM S, et al. A reconfigurable 1T1C eDRAM-based spiking neural network computing-in-memory processor for high system-level efficiency[C]//2023 IEEE International Symposium on Circuits and Systems (ISCAS). Piscataway: IEEE, 2023: 1-5.

[18] XIE S S, NI C, SAYAL A, et al. 16.2 eDRAM-CIM: Compute-in-memory design with reconfigurable embedded-dynamic-memory array realizing adaptive data converters and charge-domain computing[C]//2021 IEEE International Solid-State Circuits Conference (ISSCC). Piscataway: IEEE, 2021: 248-250.

[19] ZHAO Y S, SHEN Z X, XU J R, et al. A novel transpose 2T-DRAM based computing-in-memory architecture for on-chip DNN training and inference[C]//2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS). Piscataway: IEEE, 2023: 1-4.

[20] YU C S, YOO T, KIM H, et al. A logic-compatible eDRAM compute-in-memory with embedded ADCs for processing neural networks[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2021, 68(2): 667-679.

[21] JUN Z. Flash memory technology development[C]//2001 6th International Conference on Solid-State and Integrated Circuit Technology. Proceedings (Cat. No.01EX443). Piscataway: IEEE, 2001: 189-194.

[22] HAN R Z, XIANG Y C, HUANG P, et al. Flash memory array for efficient implementation of deep neural networks [J]. Advanced Intelligent Systems, 2021, 3(5): 2000161.

[23] LI X Y, WU H Q, GAO B, et al. Electrode-induced digital-to-analog resistive switching in TaOx-based RRAM devices [J]. Nanotechnology, 2016, 27(30): 305201.

[24] WONG H S P, RAOUX S, KIM S, et al. Phase change memory[J]. Proceedings of the IEEE, 2010, 98(12): 2201-2227.

[25] APALKOV D, DIENY B, SLAUGHTER J M. Magnetoresistive random access memory[J]. Proceedings of the IEEE, 2016, 104(10): 1796-1830.

[26] ZHANG Y Z, WU H Q, QIAN H, et al. An improved RRAM-based binarized neural network with high variation-tolerated forward/backward propagation module[J]. IEEE Transactions on Electron Devices, 2020, 67(2): 469-473.

[27] YAO P, WU H Q, GAO B, et al. Fully hardware-implemented memristor convolutional neural network[J]. Nature, 2020, 577(7792): 641-646.

[28] KHWA W S, CHIU Y C, JHANG C J, et al. A 40-nm, 2M-cell, 8b-precision, hybrid SLC-MLC PCM computing-in-memory macro with 20.5 - 65.0TOPS/W for tiny-Al edge devices[C]//2022 IEEE International Solid-State Circuits Conference (ISSCC). Piscataway: IEEE, 2022: 1-3.

[29] FREDEMAN G, PLASS D, MATHEWS A, et al. 17.4 A 14nm 1.1Mb embedded DRAM macro with 1ns access[C]//2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers. Piscataway: IEEE, 2015: 1-3.

[30] SEBASTIAN A, LE GALLO M, KHADDAM-ALJAMEH R, et al. Memory devices and applications for in-memory computing[J]. Nature Nanotechnology, 2020, 15: 529-544.

[31] WONG H S P, LEE H Y, YU S M, et al. Metal-oxide RRAM [J]. Proceedings of the IEEE, 2012, 100(6): 1951-1970.

[32] FONG X, KIM Y, VENKATESAN R, et al. Spin-transfer torque memories: Devices, circuits, and systems[J]. Proceedings of the IEEE, 2016, 104(7): 1449-1488.

[33] CHIU Y C, YANG C S, TENG S H, et al. A 22nm 4Mb STT-MRAM data-encrypted near-memory computation macro with a 192GB/s read-and-decryption bandwidth and 25.1-55.1TOPS/W 8b MAC for AI operations[C]//2022 IEEE International Solid- State Circuits Conference (ISSCC). Piscataway: IEEE, 2022: 178-180.

[34] JEONG S, PARK J, JEON D. A 28nm 1.644TFLOPS/W floating-point computation SRAM macro with variable

precision for deep neural network inference and training[C]// ESSCIRC 2022 IEEE 48th European Solid State Circuits Conference (ESSCIRC). Piscataway: IEEE, 2022: 145-148.

[35] TU F B, WANG Y Q, WU Z H, et al. A 28nm 29.2TFLOPS/W BF16 and 36.5TOPS/W INT8 reconfigurable digital CIM processor with unified FP/INT pipeline and bitwise In-memory booth multiplication for cloud deep learning acceleration[C]//2022 IEEE International Solid-State Circuits Conference (ISSCC). Piscataway: IEEE, 2022: 1-3.

[36] LIU Q, GAO B, YAO P, et al. 33.2 A fully integrated analog ReRAM based 78.4TOPS/W compute-In-memory chip with fully parallel MAC computing[C]//2020 IEEE International Solid-State Circuits Conference - (ISSCC). Piscataway: IEEE, 2020: 500-502.

[37] HSIEH S E, WEI C H, XUE C X, et al. 7.6 A 70.85-86.27TOPS/W PVT-insensitive 8b word-wise ACIM with post-processing relaxation[C]//2023 IEEE International Solid-State Circuits Conference (ISSCC). Piscataway: IEEE, 2023: 136-138.

[38] CHEN P Y, WU M, ZHAO W T, et al. 7.8 A 22nm delta-sigma computing-in-memory ($\Delta\sum$CIM) SRAM macro with near-zero-mean outputs and LSB-first ADCs achieving 21.38TOPS/W for 8b-MAC edge AI processing[C]//2023 IEEE International Solid-State Circuits Conference (ISSCC). Piscataway: IEEE, 2023: 140-142.

[39] HUNG J M, HUANG Y H, HUANG S P, et al. An 8-Mb DC-current-free binary-to-8b precision ReRAM nonvolatile computing-in-memory macro using time-space-readout with 1286.4-21.6TOPS/W for edge-AI Devices[C]//2022 IEEE International Solid-State Circuits Conference (ISSCC). Piscataway: IEEE, 2022: 1-3.

[40] BISWAS A, CHANDRAKASAN A P. Conv-RAM: An energy-efficient SRAM with embedded convolution computation for low-power CNN-based machine learning applications[C]//2018 IEEE International Solid -State Circuits Conference - (ISSCC). Piscataway: IEEE, 2018: 488-490.

[41] CHEN W H, LI K X, LIN W Y, et al. A 65nm 1Mb nonvolatile computing-in-memory ReRAM macro with sub-16ns multiply-and-accumulate for binary DNN AI edge processors[C]//2018 IEEE International Solid-State Circuits Conference - (ISSCC). Piscataway: IEEE, 2018: 494-496.

[42] XUE C X, CHEN W H, LIU J S, et al. 24.1 A 1Mb multibit ReRAM computing-In-memory macro with 14.6ns parallel MAC computing time for CNN based AI edge processors [C]//2019 IEEE International Solid-State Circuits Conference - (ISSCC). Piscataway: IEEE, 2019: 388-390.

[43] SI X, CHEN J J, TU Y N, et al. 24.5 A twin-8T SRAM computation-in-memory macro for multiple-bit CNN-based machine learning[C]//2019 IEEE International Solid-State Circuits Conference - (ISSCC). Piscataway: IEEE, 2019: 396-398.

[44] YANG J, KONG Y Y, WANG Z, et al. 24.4 sandwich-RAM: An energy-efficient in-memory BWN architecture with pulse-width modulation[C]//2019 IEEE International Solid-State Circuits Conference - (ISSCC). Piscataway: IEEE, 2019: 394-396.

[45] SU J W, SI X, CHOU Y C, et al. 15.2 A 28nm 64Kb inference-training two-way transpose multibit 6T SRAM compute-in-memory macro for AI edge chips[C]//2020 IEEE International Solid-State Circuits Conference - (ISSCC). Piscataway: IEEE, 2020: 240-242.

[46] XUE C X, HUANG T Y, LIU J S, et al. 15.4 A 22nm 2Mb ReRAM compute-in-memory macro with 121-28TOPS/W for multibit MAC computing for tiny AI edge devices[C]// 2020 IEEE International Solid-State Circuits Conference - (ISSCC). Piscataway: IEEE, 2020: 244-246.

[47] XUE C X, HUNG J M, KAO H Y, et al. 16.1 A 22nm 4Mb 8b-precision ReRAM computing-in-memory macro with 11.91 to 195.7TOPS/W for tiny AI edge devices[C]//2021 IEEE International Solid-State Circuits Conference (ISSCC). Piscataway: IEEE, 2021: 245-247.

[48] SU J W, CHOU Y C, LIU R H, et al. 16.3 A 28nm 384Kb 6T-SRAM computation-in-memory macro with 8b precision for AI edge chips[C]//2021 IEEE International Solid-State Circuits Conference (ISSCC). Piscataway: IEEE, 2021: 250-252.

[49] YAN B N, HSU J L, YU P C, et al. A 1.041-mb/mm2 27.38-TOPS/W signed-INT8 dynamic-logic-based ADC-less SRAM compute-in-memory macro in 28nm with reconfigurable bitwise operation for AI and embedded applications[C]//2022 IEEE International Solid-State Circuits Conference (ISSCC). Piscataway: IEEE, 2022: 188-190.

[50] GONUGONDLA S K, KANG M G, SHANBHAG N. A 42pJ/decision 3.12TOPS/W robust in-memory machine learning classifier with on-chip training[C]//2018 IEEE International Solid-State Circuits Conference - (ISSCC). Piscataway: IEEE, 2018: 490-492.

[51] JHANG C J, XUE C X, HUNG J M, et al. Challenges and trends of SRAM-based computing-in-memory for AI edge devices[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2021, 68(5): 1773-1786.

[52] GUO R Q, LIU Y G, ZHENG S X, et al. A 5.1pJ/Neuron

127.3us/Inference RNN-based speech recognition processor using 16 computing-in-memory SRAM macros in 65nm CMOS[C]//2019 Symposium on VLSI Circuits. Piscataway: IEEE, 2019: C120-C121.

[53] AGRAWAL A, KOSTA A, KODGE S, et al. CASH-RAM: Enabling in-memory computations for edge inference using charge accumulation and sharing in standard 8T-SRAM arrays[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2020, 10(3): 295-305.

[54] YUE J S, YUAN Z, FENG X Y, et al. 14.3 A 65nm computing-in-memory-based CNN processor with 2.9-to-35.8TOPS/W system energy efficiency using dynamic-sparsity performance-scaling architecture and energy-efficient inter/intra-macro data reuse[C]//2020 IEEE International Solid-State Circuits Conference - (ISSCC). Piscataway: IEEE, 2020: 234-236.

[55] NGUYEN V T, KIM J S, LEE J W. 10T SRAM computing-in-memory macros for binary and multibit MAC operation of DNN edge processors[J]. IEEE Access, 2021, 9: 71262-71276.

[56] SU J W, SI X, CHOU Y C, et al. Two-way transpose multibit 6T SRAM computing-in-memory macro for inference-training AI edge chips[J]. IEEE Journal of Solid-State Circuits, 2022, 57(2): 609-624.

[57] BISHNOI R, EBRAHIMI M, OBORIL F, et al. Read disturb fault detection in STT-MRAM[C]//2014 International Test Conference. Piscataway: IEEE, 2014: 1-7.

[58] PIROVANO A, REDAELLI A, PELLIZZER F, et al. Reliability study of phase-change nonvolatile memories[J]. IEEE Transactions on Device and Materials Reliability, 2004, 4(3): 422-427.

[59] CAI Y, LUO Y X, GHOSE S, et al. Read disturb errors in MLC NAND flash memory: Characterization, mitigation, and recovery[C]//2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. Piscataway: IEEE, 2015: 438-449.

[60] SI X, TU Y N, HUANG W H, et al. A Local Computing Cell and 6T SRAM-Based Computing-in-Memory Macro With 8-b MAC Operation for Edge AI Chips[J]. IEEE Journal of Solid-State Circuits, 2021, 56(9): 2817-2831.

[61] LI H T, JIANG Z Z, HUANG P, et al. Statistical assessment methodology for the design and optimization of cross-point RRAM arrays[C]//2014 IEEE 6th International Memory Workshop (IMW). Piscataway: IEEE, 2014: 1-4.

[62] YU S M, SHIM W, PENG X C, et al. RRAM for compute-in-memory: From inference to training[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2021, 68(7): 2753-2765.

[63] YIN S H, JIANG Z W, SEO J S, et al. XNOR-SRAM: In-memory computing SRAM macro for binary/ternary deep neural networks[J]. IEEE Journal of Solid-State Circuits, 2020: 1-11.

[64] KIM Y, KIM H, PARK J, et al. Mapping binary ResNets on computing-in-memory hardware with low-bit ADCs[C]//2021 Design, Automation & Test in Europe Conference & Exhibition (DATE). Piscataway: IEEE, 2021: 856-861.

[65] WANG H C, LIU R Z, DORRANCE R, et al. A charge domain SRAM compute-in-memory macro with C-2C ladder-based 8-bit MAC unit in 22-nm FinFET process for edge inference[J]. IEEE Journal of Solid-State Circuits, 2023, 58(4): 1037-1050.

[66] YU S M, JIANG H W, HUANG S S, et al. Compute-in-memory chips for deep learning: Recent trends and prospects [J]. IEEE Circuits and Systems Magazine, 2021, 21(3): 31-56.

[67] SHIM W, MENG J, PENG X C, et al. Impact of multilevel retention characteristics on RRAM based DNN inference engine[C]//2021 IEEE International Reliability Physics Symposium (IRPS). Piscataway: IEEE, 2021: 1-4.

[68] DONG Q, WANG Z H, LIM J, et al. A 1Mb 28nm STT-MRAM with 2.8ns read access time at 1.2V VDD using single-cap offset-cancelled sense amplifier and in situ self-write-termination[C]//2018 IEEE International Solid - State Circuits Conference - (ISSCC). Piscataway: IEEE, 2018: 480-482.

[69] AMBROGIO S, NARAYANAN P, OKAZAKI A, et al. An analog-AI chip for energy-efficient speech recognition and transcription[J]. Nature, 2023, 620(7975): 768-775.

[70] NABAVINEJAD S M, BAHARLOO M, CHEN K C, et al. An overview of efficient interconnection networks for deep neural network accelerators[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2020, 10(3): 268-282.

[71] BJERREGAARD T, MAHADEVAN S. A survey of research and practices of network-on-chip[J]. ACM, 2006, 38(1): 1-51.

[72] HEMMATI M R, DOLATSHAHI M, MEHRBOD A. Increasing the efficiency of NOC routing algorithms based on fault tolerance measurement method[C]//2018 International Young Engineers Forum (YEF-ECE). Piscataway: IEEE, 2018: 31-38.

[73] SMAGULOVA K, FOUDA M E, KURDAHI F, et al. Resistive neural hardware accelerators[J]. Proceedings of the

IEEE, 2023, 111(5): 500-527.
[74] ZHANG W Q, GAO B, TANG J S, et al. Neuro-inspired computing chips[J]. Nature Electronics, 2020, 3: 371-382.
[75] FEI X, ZHANG Y H, ZHENG W M. XB-SIM: A simulation framework for modeling and exploration of ReRAM-based CNN acceleration design[J]. Tsinghua Science and Technology, 2021, 26(3): 322-334.
[76] KRISHNAN G, MANDAL S K, CHAKRABARTI C, et al. System-level benchmarking of chiplet-based IMC architectures for deep neural network acceleration[C]//2021 IEEE 14th International Conference on ASIC (ASICON). Piscataway: IEEE, 2021: 1-4.

作者简介

李嘉宁　男,2021年于吉林大学电子科学与工程学院获学士学位,2021年至今于清华大学集成电路学院攻读博士学位. 目前的主要研究方向为存算一体电路与架构.

姚　鹏　男,2014年于西安交通大学获微电子学学士学位,2020年于清华大学获博士学位. 已在《自然》、《科学》、《自然通讯》、*ISSCC*、*IEDM*和*VLSI*等期刊和会议上发表或合作发表了多篇论文. 主要研究方向:存算一体技术、神经形态计算等.

揭　路　男,2013-2017年于浙江大学信息与电子工程学院获学士学位, 2021年于美国密歇根大学电气与计算机工程系获博士学位. 现为集成电路学院助理教授. 研究方向:模数/数模转换器、可重构数模混合电路、数模混合计算等.

唐建石　男,2008年本科毕业于清华大学微纳电子系,2014年博士毕业于美国UCLA电子工程系,2015-2019年在美国IBM T. J. Watson Research Center工作,2019年回清华大学工作,现任清华大学集成电路学院副教授. 主要研究方向包括新型存储器与类脑计算、单片三维异质集成等. 中国电子学会会员编号:E190034508M.

伍　冬　男,2001年7月毕业于西安交通大学电子工程系,获学士学位,2006年7月毕业于清华大学微电子学研究所,获博士学位,现为集成电路学院副研究员. 研究方向:主要从事图像传感器和非挥发性存储器等阵列式电路系统设计技术研究.

高　滨　男,2008年和2013年分别在北京大学获得物理学学士学位和微电子学博士学位. 目前,目前在清华大学集成电路学院担任副教授. 主要研究方向为新型半导体器件的制造、表征和理论建模,特别强调阻变随机存取存储器(RRAM).

钱　鹤　男,1990年毕业于西安交通大学微电子专业并获博士学位;1990年12月~2006年5月在中科院微电子所工作,并于2001年9月~2006年5月任该所所长;2006年6月~2008年12月在三星半导体(中国)研究所工作,任所长;2009年1月起入职清华大学. 科研工作主要集中在新型半导体存储器方面,包括面向嵌入式存储和安全认证应用,以及基于忆阻器的存算一体(CIM)芯片研发等.

吴华强　男,2000年毕业于清华大学材料科学与工程系,获得工学学士学位;同年获清华大学经济管理学院管理学士学位(双学位). 2005年在美国康奈尔大学(Cornell University)电子与计算机工程学院获工学博士学位. 随后先后在美国Spansion公司和美国Primet Precision Materials公司分别担任高级工程师和技术主管. 2009年,加入清华大学微电子学研究所,现任清华大学集成电路学院院长. 长期从事新型存储器及基于忆阻器的存算一体研究,涵盖了从器件、工艺集成、架构、算法、芯片以及系统等多个层次. 中国电子学会会员编号:E190085238M.
E-mail: wuhq@tsinghua.edu.cn