**存内计算与AI芯片设计综述**

**摘要**

冯诺依曼结构存在着存储墙问题。近年来，传统的计算机体系结构也面临着严峻的性能挑战。人工智能应用中需要大量的数据传输与计算。存内计算通过赋予内存一定的计算能力，有望缓解AI芯片中的内存墙的瓶颈。在本文中，我们介绍了存内计算思想的产生背景，原理，调研了目前的存内计算AI芯片和存内计算芯片市场需求，基于调研的内容，给出我们对于存内计算芯片设计未来的展望。

1. **存内计算概述**

近年来，传统的计算机体系结构面临着严峻的性能和能效挑战。在过去的几十年里，处理器的性能遵循摩尔定律发展，每年提高60%，而存储器的性能每年只提高7%[1]。由于传统的计算机系统中计算和存储的分离，处理器计算时，数据需要从内存中搬运。存储器和处理器之间的数据运输造成了大量的数据移动能耗。为了缓解访存延迟的问题，传统的计算机系统结构通过层次化存储结构，利用应用所具有的数据局部性，设计和优化缓存机制，降低访存带来的延迟。然而，近年来，大量的人工智能应用呈现了新的数据特点：访存局部性差和计算访存比低。访存局部性差，导致缓存的命中率低。计算访存比低，计算系统的大部分时间用于访存，数据搬运占用了大部分时间。因此，传统的基于缓存机制的计算机架构在处理人工智能应用时数据传输效率低下。

存内计算的设计思想，是试图使计算离存储器更近。存内计算通过赋予存储器一定的计算能力，在存储器上实现部分的计算功能，降低数据在存储器和处理器之间的搬运产生的能耗，提升计算系统的性能。存内计算有望在一定程度上缓解内存墙的瓶颈。通过将一些轻量级的计算单元集成到内存中，存内计算方式可以减轻处理器和存储器之间的数据传输压力。存内计算的思想最早可以追溯到上世纪七十年代[2]。然而，存内计算计算机系统的真正意义上的流片实现却要到九十年代。受限于当时的技术水平，存内计算思想在上世纪并没有发展起来。当前，传统的基于缓存机制的计算机系统在处理一些计算访存比低的人工智能应用时面临着缓存命中率低，数据搬运能耗高的问题。存内计算架构有望提高这些应用的处理效率。三维堆叠技术和新兴的非易失性存储器件为存内计算的实现提供了技术基础和器件基础。存内计算机的实现主要有两类。第一类存内计算机的实现，利用了三维堆叠内存技术[3,4,5]。第二类存内计算机的实现，采用了新兴的非易失性存储设备，并且修改了存内设计，以实现存储器中的逻辑运算。近年来，人们相继提出了基于各种新兴的非易失性存储器件的存内计算机架构，如以自旋传递转矩存储器STT-RAMs为存储单元的存内计算机架构[6]，以阻变存储器RRAMs为存储器件的存内计算机架构[7,8]，以相变存储器PCMs为存储元件的存内计算机架构[9]，以及以铁电场效应晶体管FeFETs为存储器件的存内计算机架构[10]。

1. **存内计算实现原理**
2. 三维堆叠技术
3. 非易失存储器
4. STTRAM
5. RRAMs
6. PCMs
7. FeFET
8. **存内计算AI芯片**
9. **市场分析**

存内计算虽然可以突破传统冯诺依曼架构的瓶颈，但是仍然面临若干技术瓶颈，如硬件资源的复用、存算一体化单元的设计和模拟量运算的实现等问题。

传统冯诺依曼架构将存储单元和逻辑单元分离开来，并根据距离逻辑单元的远近，将存储单元划分为硬盘、内存、缓存等多级存储模式。这使得有限的逻辑单元可以调用大量的存储信息，实现了逻辑单元的复用，从而降低了片上资源的开销。然而在存内计算架构中，存储单元和逻辑单元关系发生改变，主要存在3种模式：存储单元分散在逻辑单元内、逻辑单元分散在存储单元内或二者有机融合。这些都将导致单位逻辑单元可调用的存储信息规模的下降，从而降低了逻辑单元的复用能力，增加片上开销。网络层数是当前人工智能算法的一个重要指标，直接影响着算法性能。由于存在硬件资源复用的问题，层数的增加对存内计算架构的影响远大于对传统架构的影响，从而限制了存内计算的应用场景。

存内运算对存算一体化单元的性能要求十分苛刻。在开销上，存算一体化单元的硬件开销应该控制在一定范围内，至少要明显低于分立的存储单元和逻辑单元的开销总和。例如，基于CMOS电路的存内计算单元的硬件开销只有逐步趋近SRAM的量级，才能体现存内计算架构的存在意义。在可靠性上，存算一体化单元应该同时满足存储和计算两者的可靠性需求，如存储单元的循环特性、保持特性、一致性等以及逻辑单元的响应速度、功耗等。然而，当前的半导体技术，主要还是围绕存储、逻辑模块分别进行针对性优化。从器件工艺和电路设计上，同时照顾到二者的协同优化仍是挑战。

人工智能算法的运算涉及批量的准模拟量运算，例如，算法对权重的变化范围和精度要求较为苛刻，尤其是涉及网络的训练过程，权重精度的要求将超过6bits。然而，在硬件实现过程中，数字量的精准运算开销较大。反而，具备模拟量功能的趋势性估算既能够简化算法，又能够降低对硬件的性能要求。当前，逻辑单元设计主要是基于数字量的精准运算，模拟量运算设计的理论和经验相对缺乏。另外，在硬件上实现模拟量的可控调制和存储的难度较高，也是模拟量运算难以实现的原因之一。

目前，将存内计算作为人工智能加速的方案相比于传统FPGA和GPU等方案还不够成熟，但其未来的发展方向较为明确。随着人工智能的硬件化加速方法的不断发展，多元化的加速方法层出不穷，主要可分为云端训练、云端推理和终端推理三大类。终端训练的技术难度较高，故在市场上产品迭代速度较慢，但却是存内计算的主要发展方向。终端推理过程涉及的权重精度可以进一步简化，降低计算复杂度，从而更利于存储和计算；另外，终端推理过程不涉及权重的更新，无需对存内信息进行频繁修正，从而降低了存算一体化单元的性能要求。相较于云端，终端所涉及的任务较为固定，对硬件的加速能力要求不高，无需频繁变动架构，更适合实现存内计算。

存内计算在物联网方面具有显著的应用潜力。物联网涉及大量的智能终端，其中的计算过程往往包含人工智能的数据预处理。存内计算由于硬件复用问题，在深层算法的加速方面不占优势，反而在浅层算法的实现过程中，简化的存算结构更易于硬件实现。由于存内计算无需频繁搬用数据，能耗将显著降低，符合物联网对硬件低能耗的要求，大大提高物联网的计算效率。

为了加速存内计算应用于实际场景，应重点提高非挥发存储器的计算能力。基于SRAM框架实现的存算一体设计，其模拟特性、集成能力等都不符合物联网终端的发展趋势。而非挥发存储器件，如NOR Flash、RRAM等，具有模拟量存储、非挥发、低功耗等其显著优势，体现了存内计算方案核心竞争力。在非挥发存储器范畴内，以NOR Flash作为存储器的技术较为成熟，工艺参数、器件模型和模块设计均拥有成熟工具，因此其存内计算解决方案将最先实现。未来，NOR Flash的系统架构的核心设计或许可以向RRAM等新型非挥发器件迁移，从而能够实现技术迭代革新和产品延续，具有较高的商业潜力。

目前存算一体的AI芯片主要应用于音频类、健康类及低功耗视觉端侧应用场景中。在语音识别领域，存算一体AI芯片不仅能实现语音唤醒功能，并支持通话降噪、声纹识别等功能。此外，健康领域应用也可以通过存算一体芯片实现，由于健康类算法的运算量比语音类更小，因此芯片功耗相应更低。自2016年UCSB发布存算一体AI芯片第一个样片以来，多家国内外初创企业陆续开展相关研发，如美国Mythic、Syntiant、中国知存科技和闪亿半导体等，并得到包括Intel、ARM、Bosch、Amazon、Microsoft、Softbank、中芯国际等国内外主流半导体企业与资本的产业投资。从下游市场空间来看，存内计算的市场前景将持续向好。

1. **总结**

参考文献

[1] D. Patterson et al. 1997. A case for intelligent RAM. IEEE Micro 17, 2 (March

1997), 34–44

[2]WILLIAM H. KAUTZ,1969 Cellular logic-In-Memory Arrays, IEEE Trans. Computers, vol. pp.719-727

[3]J. Ahn et al. 2015. A scalable Processing-in-Memory accelerator for parallel graph

processing. In ISCA. 105–117.

[4]M. Gao et al. 2017. TETRIS: Scalable and Efficient Neural Network Acceleration

with 3D Memory. In ASPLOS (ASPLOS ’17). 751–764

[5] S. H. Pugsley et al. 2014. NDC: Analyzing the impact of 3D-stacked memory+logic devices on MapReduce workloads. In ISPASS. 190–200

[6] S. Jain et al. 2018. Computing in Memory With Spin-Transfer Torque Magnetic

RAM. IEEE TVLSI 26, 3 (March 2018), 470–483.

1. P. Gaillardon et al. 2016. The Programmable Logic-in-Memory (PLiM) computer. In DATE. 427–432.
2. M. Imani et al. 2017. MPIM: Multi-purpose in-memory processing using configurable resistive memory. In ASP-DAC. 757–763
3. S. Li et al. 2016. Pinatubo: A Processing-in-Memory architecture for bulk bitwise operations in emerging non-volatile memories. In DAC. 1–6.

[10]D. Reis et al. 2018. Computing in Memory with FeFETs. In ISLPED (ISLPED ’18).24:1–24:6.