物理学报 Acta Physica Sinica

Chinese Physical Society

Institute of Physics, CAS

基于非挥发存储器的存内计算技术

周正　黄鹏　康晋锋

Non-volatile memory based in-memory computing technology

Zhou Zheng　Huang Peng　Kang Jin-Feng

引用信息 Citation: Acta Physica Sinica, 71, 148507 (2022)　DOI: 10.7498/aps.71.20220397

在线阅读 View online: https://doi.org/10.7498/aps.71.20220397

当期内容 View table of contents: http://wulixb.iphy.ac.cn

专题: 面向类脑计算的物理电子学

基于非挥发存储器的存内计算技术

周正 黄鹏 康晋锋†

(北京大学集成电路学院, 北京 100871)

(2022 年 3 月 5 日收到; 2022 年 6 月 10 日收到修改稿)

通过在基本单元上集成存储和计算功能, 存内计算技术能够显著降低数据搬运规模, 被广泛认为是突破传统冯·诺依曼计算架构性能瓶颈的新型计算范式. 非挥发存储器件兼具非易失特性和存算融合功能, 是实现存内计算的良好功能器件. 本文首先介绍了存内计算范式的基本概念, 包括技术背景和技术特征. 然后综述了用于实现存内计算的非挥发存储器件及其性能特征, 包含传统闪存器件和新型阻变存储器; 进一步介绍了基于非挥发存储器件的存内计算实现方法, 包括存内模拟运算和存内数字运算. 之后综述了非挥发存内计算系统在深度学习硬件加速、类脑计算等领域的潜在应用. 最后, 对非挥发型存内计算技术的未来发展趋势进行了总结和展望.

关键词: 存内计算, 非挥发存储器, 闪存, 阻变存储器

PACS: 85.35.–p, 07.05.Mh, 84.35.+i, 85.30.Tv **DOI**: 10.7498/aps.71.20220397

1 引 言

随着集成电路技术的发展, 电子设备逐渐推广到日常生活的各个方面[1]. 电子设备互联互通, 产生的数模规模不断攀升. 预计 2025 年, 全球的实时数据规模将达到 47 泽字节 (1 泽字节 = 2^{70} 字节)[2]. 由此催生出众多数据依赖的信息处理任务, 如模式识别、数字孪生等. 然而, 由于摩尔定律接近物理极限, 集成电路等比例缩小趋势逐步放缓[3], 传统计算平台性能进一步提升面临严峻挑战. 同时, 传统计算平台受限于冯·诺依曼瓶颈[4] 和存储墙[5] 等问题, 难以应对海量数据搬运和处理的实际需求. 存内计算技术通过在基本单元上集成计算和存储功能, 打破了传统冯·诺依曼瓶颈, 能够显著降低数据搬运, 被认为是未来计算架构的重要发展趋势之一[6–8].

非挥发存储 (non-volatile memory, NVM) 器件是实现存内计算的优良器件. 首先, NVM 器件具备断电数据保持特性, 工作状态下无需预加载数据, 待机状态下无需多余的能耗开销, 从而具备低功耗特性; 其次, NVM 器件具备多值/模拟存储特性[9], 能够实现高密度的数据存储和信息处理能力; 最后, NVM 器件能够在器件层次上集成存储和计算功能[10], 实现存内计算的基本单元结构简单、集成度高, 具备良好的等比例缩小能力[11]. NVM 器件根据技术成熟度, 可分为传统 NVM 器件和新型 NVM 器件[12]. 闪存 (flash) 器件是典型的传统 NVM 器件, 具备工艺成熟、性能稳定等优势. 研究人员拓展了传统 flash 器件的存内计算功能, 实现了诸如卷积神经网络加速[13] 和稳态逻辑计算[14] 等应用. 新型 NVM 器件类型众多, 具备高集成度、低功耗和响应迅速等预期优势, 被认为能够在存储金字塔中弥补传统存储和内存之间的存储级内存[15]. 进一步地, 新型 NVM 器件表现出优良模拟双向调制特性[16], 具备高密度存内计算应用潜力.

基于 NVM 器件, 研究人员拓展出多种存内计算模式. 根据运算类型, 可分为存内模拟运算和存

† 通信作者. E-mail: kangjf@pku.edu.cn

内数字运算[17]. 存内模拟运算主要利用 NVM 器件及其阵列结构的器件响应特性和信号调制能力进行运算, 在深度学习加速[18] 以及线性方程组求解[19] 等方面具备显著优势. 存内数字运算则利用多个 NVM 器件的存储状态, 通过激励信号和器件之间的相互作用进行逻辑运算, 形成了随机计算和布尔逻辑[20] 等发展方向. 总体来看, 近年来随着非挥发型存内计算技术研究工作的不断深入, 逐步形成了非挥发存内计算器件功能开发、存内计算运算模式设计实现和存内计算系统应用开发逐步递进的研究体系. 本文将聚焦在非挥发存内计算技术路线, 从 NVM 器件、存内计算运算模式和存内计算应用场景三个层面, 回顾近年来研究人员在非挥发型存内计算技术领域所取得的最新研究进展, 总结当前面临的关键问题, 展望未来发展趋势与前景.

2 存内计算范式

根据存储模块与计算模块的相对关系, 可以将硬件系统的计算范式分为存算分离范式、近存计算范式和存内计算范式 (又称为“存算一体”或“存算融合”)[21,22], 如图 1 所示. 传统冯·诺依曼架构不仅是典型的存算分离的计算范式, 还是当前主流处理器 CPU 和 GPU 等计算平台的架构基础. 其特征是计算部分与存储部分相互独立, 并通过总线连接[23], 具备“程序存储、共享数据、顺序执行”的特点; 此外以计算单元为中心执行任务, 严重依赖存储元件的数据交互能力. 可把近存计算范式看作是对存算分离范式的优化设计. 通过平衡系统中存储体系各部分速度和容量、引入高带宽存储模块、增加片上存储容量等手段, 降低数据搬运延时、提高数据带宽, 从而提升系统性能. 张量处理器 (tensor processing unit, TPU)[24] 和网络处理器 (neural-network process unit, NPU)[25] 等新型计算平台是近存计算的典型代表. 存内计算范式[26] 则从根本上改变了存储和计算的关系, 其特征是在基本单元内同时实现计算和存储功能, 模糊了存储和计算的界限, 从根本上缓解了数据搬运的问题.

存内计算作为一种新型计算范式, 对硬件单元、运算逻辑和系统架构都提出了全新的需求和挑战, 并需要针对应用场景进行定制化开发. 首先, 存内计算基本单元是构成存算融合的基础, 需同时具备存储和计算两种功能. 存内计算将基本单元的存储状态视为逻辑计算的输入或输出, 在基本单元内部或附近实现计算功能, 由此减少输入或输出的数据搬运. 构成存内计算基本单元的可以是复杂的逻辑电路[27], 也可能是器件组合[28] 甚至单个器件[29]. 从集成角度看, 单个器件或器件组合能够显著降低存内计算单元的开销. 其次, 在基本单元的基础上需构建恰当的运算逻辑. 与互补金属氧化物半导体 (complementary metal-oxide-semiconductor, CMOS) 逻辑门不同, 存内计算不再局限在布尔逻辑范畴内, 而是适当地引入硬件兼容的算数运算作为基本运算功能, 如加法、乘法等, 从而在有限的硬件资源条件下, 提高特定任务的计算效率. 最后, 受输入输出数据的形式和基本运算类型的影响, 存内计算的系统架构面临诸多挑战. 诸如算数运算的输入输出模拟信号处理问题[30]、运算过程的级联和中间数据的存储问题[31] 以及硬件资源复用问题[32] 等. 针对实际任务进行定向开发是存内计算技术的重要推动力量. 因此, 存内计算的研究工作, 往往是根据任务的运算需求和数据形式, 如深度学习的多层网络结构、状态逻辑的级联特点等, 进行定制化的基本单元设计和优化, 进而研究基本运算的实现方式, 提出对应的存内计算系统架构.

3 非挥发型存算一体功能器件

根据数据是否断电易失, 可以将存内计算分为挥发型和非挥发型. 以静态随机存取存储器 (static

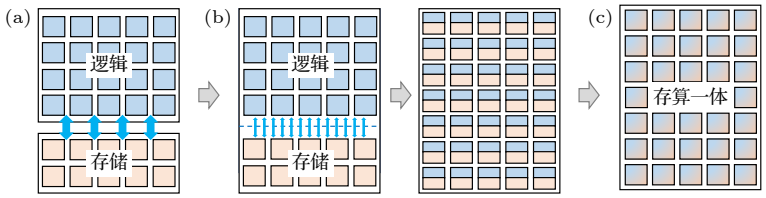

图 1 传统计算范式分类 (a) 存算分离计算; (b) 近存计算; (c) 存内计算

Fig. 1. Classification of the traditional computational paradigms: (a) Separated memory and logic computing; (b) near memory computing; (c) in-memory computing.

random-access memory, SRAM) 为代表的易失器件可以构成典型的易失型存内计算系统[33,34]. 该系统具备配置灵活、性能稳定和技术成熟等优势, 能够显著缓解存储墙等问题. 然而, 由于缺乏断电数据保持特性, 需要进行初始化数据搬运, 从而无法脱离存储模块而独立工作. 同时, 基于 SRAM 的存算融合基本单元结构复杂, 超大规模集成存在挑战. 非易失型存算融合系统主要依托于非易失型存算一体功能器件, 包括以 flash 为代表的传统非易失存储器和以阻变存储器 (resistive random access memory, RRAM) 为代表的新型非易失存储器件. 由于具备非易失特性, 系统运行过程不仅无需外部数据的预加载, 还具备低待机功耗、快速响应等性能优势. 另外, 由于非易失存算功能器件在器件层级上就实现了存储和计算功能的集成, 从而具备更好的等比例缩小能力, 在集成大规模系统方面具有天然优势.

3.1 传统 NVM 存算功能器件: Flash

Flash 器件具有工艺成熟、性能稳定和产业化程度高等显著优势, 是存算融合系统应用实现的优良硬件载体[35]. 如图 2(a) 所示, flash 器件的浮栅结构位于栅极氧化层和控制层之间, 由具备电子缺陷的绝缘层和两侧 SiO_2 材料层构成[36]. 电荷存储在电子缺陷层不连续的缺陷状态中, 可以通过热电子注入和 Fowler-Nordheim (F-N) 隧穿效应, 改变电荷的存储状态, 从而实现数据的写入和擦除[36], 见图 2(b). 由于需要克服界面势垒, 数据写入需要较高电压 (>10 V) 和较长时间 (>10 μs), 循环擦写次数约为 10^5—10^6 次. 根据 flash 单元的排列方式, 主流 flash 阵列结构可分为 NOR 型和 NAND 型. 在 NOR 型 flash 阵列中, 如图 2(c) 所示, 相同位线 (bit line) 的 flash 呈并联结构; NAND 结构中, 如图 2(d) 所示, flash 通过控制管与位线和源线 (source line) 相互串联[36].

基于 flash 的存内计算研究涵盖了器件、阵列和系统架构等方面. 2015 年, 加州大学圣塔芭芭拉分校 (UCSB) 课题组[37] 在商用 ESF1NOR 型 flash 的基础上开发了高精度的编程算法, 实现了高于 10 位的精确存储能力. 为了降低功耗, 利用 Flash 亚阈值区域, 限制单元电流在 10^{-10}—10^{-6}A 范围内. 2017 年, UCSB 课题组[38] 进一步实现了首款基于 180 nm NOR 型 flash 工艺的全连接神经网络加速芯片. 2018 年, 北京大学 Han 等[39] 提出基于 NOR

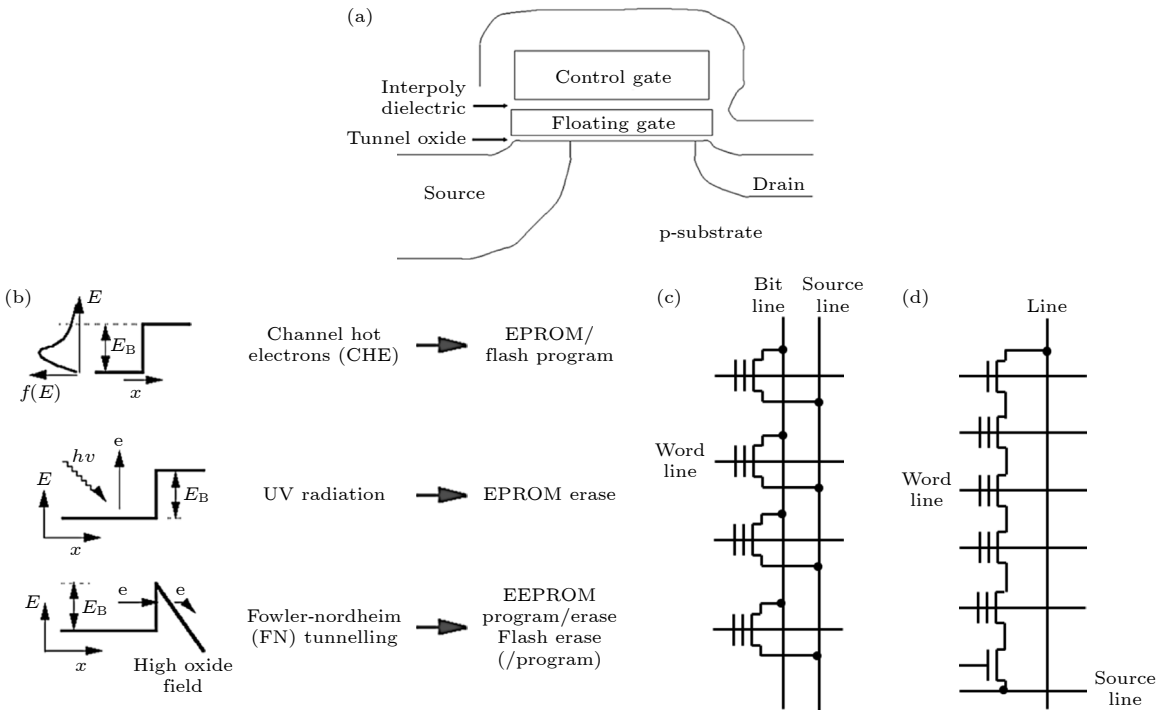

图 2 传统 NVM 器件 flash[36] (a) 典型器件结构; (b) flash 操作模式与物理机制; (c) NOR 型阵列结构; (d) NAND 型阵列结构

Fig. 2. Flash, a traditional NVM device[36]: (a) Typical flash device structure; (b) the operation scheme and physical mechanism; (c) NOR flash array; (d) NAND flash array.

型 flash 的卷积计算加速方案, 并基于 65 nm flash 工艺节点进行了流片验证, 结果表明单位功耗下处理器可执行运算操作 14.5 次, 即能效比达到 14.5. 在 2019 年, 明尼苏达大学 Kim 等 [40] 研发了基于 eNAND 型 flash 的矩阵向量乘法器, 并协同压控振荡器实现了 LeNet-5 网络. 2019 年, 旺宏公司成功演示了基于 64 GB 容量、SLC 存储密度和 3D 堆叠结构的 NAND 型 flash 颗粒, 研制出矩阵向量乘积运算加速器 [41].

3.2 新型 NVM 存算功能器件: RRAM

为了进一步提升存算融合系统性能, 相继提出了一系列新型非挥发存算功能器件. 如以相变效应形式存储信息的相变存储器 [42], 即利用电流产生的焦耳热促使相变材料在晶态和非晶态之间发生转变, 非晶态为高阻、晶态为低阻; 以自旋转移力矩形式存储数据的磁阻随机存储器 [43], 由自由磁层、隧穿层和固定磁层组成, 当自由磁层和固定磁层的磁场方向平行, 表现为低阻态, 反之为高阻态; 以自发极化铁电效应形式存储信息的铁电器件 [44], 即利用铁电材料在不同电场作用下, 晶体中原子产生位移而导致正负电荷的中心位置发生偏移, 形成极化向上和向下两种稳定状态; 以细丝导电通道通断效应存储信息的 RRAM[45]. 其中, RRAM 具备低功耗、高密度集成等优势, 是新型 NVM 器的典型代表. RRAM 结构简单, 单元面积可达 $4F^2$ (F 为光刻工艺所能达到的最小特征尺寸), 具备三维集成能力. 同时, RRAM 可实现约 85 ps 的电阻转变、擦写次数大于 10^{12}、擦写电流小于 15 μA、阻变窗口大于 100、高温 (150 ℃) 数据保持时间超过 10 年.

典型的 RRAM 工作过程可分为 3 个阶段, 如图 3(a) 所示: 1) 导线细丝初始化过程 (forming), 一般是利用较大幅度的电压或电流作用器件, 形成

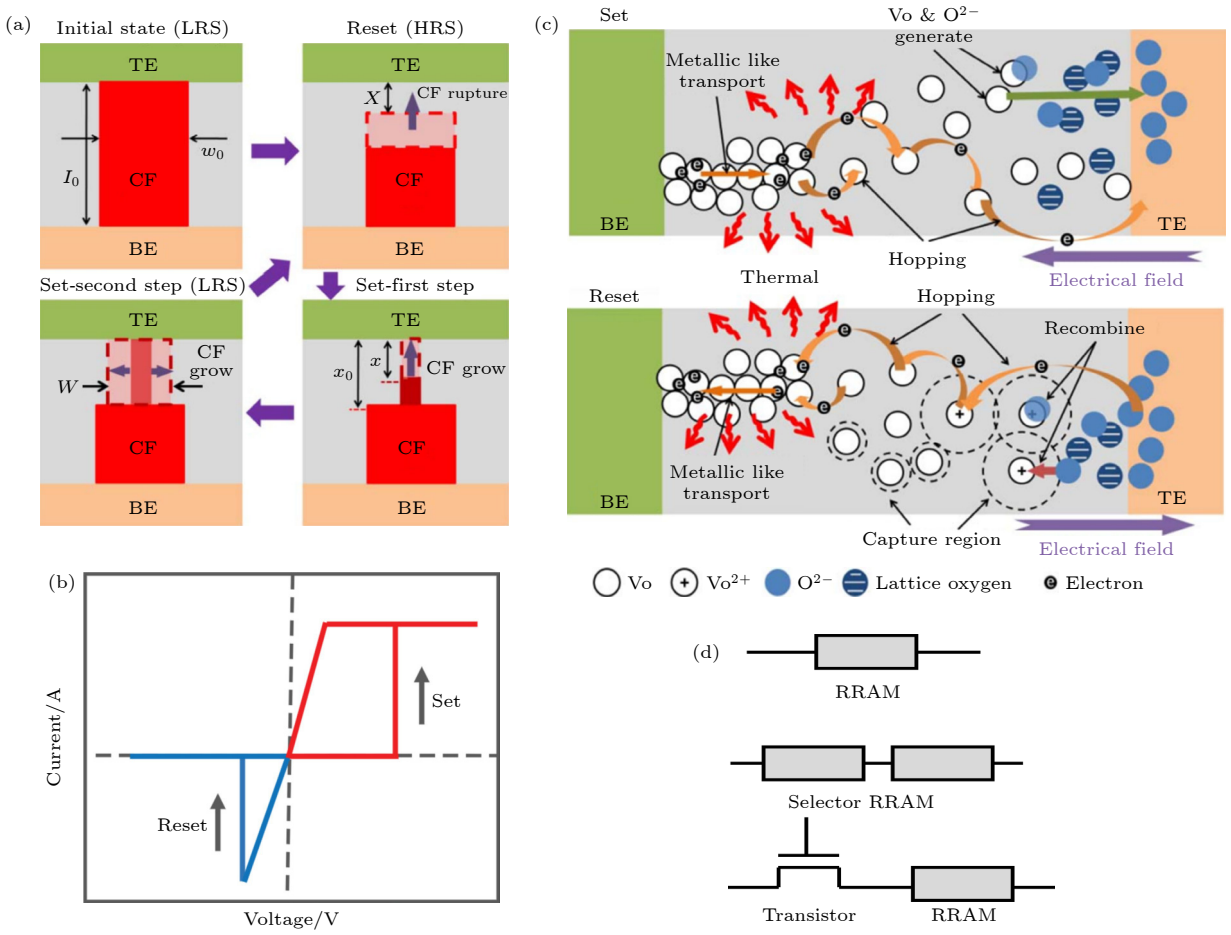

图 3 新型 NVM 器件 RRAM (a) 常见阻变行为[46]; (b) 双极型 RRAM 的典型 I-V 特性曲线; (c) 基于氧空位的阻变物理机制模型[46]; (d) 常见 RRAM 单元结构包括 1R, 1S1R 和 1T1R

Fig. 3. RRAM, a novel NVM device: (a) Typical resistive switch behavior[46]; (b) the typical I-V curve of bipolar RRAM device; (c) the physical mechanism of oxide-based RRAM[46]; (d) the typical basic unit based on RRAM include 1R, 1S1R and 1T1R.

软击穿, 使其从初始阻态不可逆地转变为低阻态; 2) 复位过程 (reset), 利用脉冲或直流信号, 在阻变器件两端形成反向电势差, 使其从低阻态转变为高阻态; 3) 置位过程 (set), 同样利用脉冲或直流信号, 在器件两端形成正向电势差, 使其从高阻态转变至低阻态. 双极型 RRAM 的典型 I-V 特性曲线中 (图 3(b)), 置位和复位是非破坏性的可逆过程[46]. 双极型阻变器件的电阻转变过程受外加电压极性的控制, 置位操作和复位操作施加的电压极性相反且方向固定. RRAM 的物理机制如图 3(c) 所示, 在初始化/置位过程外场作用下, 处于格点上的氧离子被激发为游离态, 同时在原位上留下氧空位缺陷[46]. 游离态氧离子在电场作用下逆着电场跳跃到电极和氧化物的界面处, 被电极吸附并存储在电极中. 伴随着更多的氧空位产生, 形成了一条连通上下两个电极的导电通道. 在复位过程中, 电场方向反转, 存储在电极附近的游离态氧离子被释放, 并跳跃到氧空位的俘获截面内与氧空位复合, 导致导电通道发生断裂, 器件从低阻态转变为高阻态. 常见的 RRAM 单元结构如图 3(d) 所示, 包括 1R, 1S1R 和 1T1R 结构. 1R 具备 $4F^2$ 的器件集成密度优势, 1T1R 限制了阵列的串扰能够实现较大阵列规模. 1S1R 兼具二者优势, 但面临选通管器件工艺不成熟, 1S1R 协同设计难度大等问题.

虽然在 20 世纪 60 年代已经发现了阻变现象, 但 RRAM 真正开始得到学术界和工业界的广泛关注, 是从 2004 年三星公司在国际电子器件大会上发布基于 NiO 的 RRAM 器件开始[47]. 此后, 基于不同材料体系的忆阻行为相继被发现, 包括氧化物、电解液、低维材料等无机材料体系和纳米纤维素、聚合物等有机材料体系[48–50]. 导电前段移动模型、导电细丝数量调控模型、肖特基势垒模型等物理机制模型[9] 也相继被提出用以解释阻变行为. 与此同时, 基于阻变器件的存内计算应用研究相继展开, 逐渐发展出深度学习加速、类脑计算、状态逻辑等新兴研究方向[51].

4 非挥发型存内计算模式

根据运算不同的模式, 可以将存内计算模式分为存内模拟运算和存内数字运算. 存内模拟运算是利用器件的模拟特性和信号调制能力, 结合器件阵列的结构特征, 实现如乘法、加法等基本算术运算, 从而在存内计算单元阵列上完成模拟运算. 存内数字运算的主要特点是利用固定的外界激励信号, 通过多个存内计算单元之间的相互作用和单元的存储状态, 以满足布尔逻辑的方式实现存内运算功能.

4.1 存内模拟运算

4.1.1 向量-矩阵运算模式

如图 4(a) 所示, 存内计算单元在二维空间呈阵列分布, 并通过交叉结构相互连接. 利用存内计算单元的存储特性, 可将矩阵元素映射至单元中. 在阵列的每行, 同时输入激励信号来表示输入向量. 同行存算单元同时对激励信号做出响应, 将同列存算单元的响应累积起来构成列向量输出, 即向量矩阵乘积结果. 以两端器件 RRAM 为例, 利用欧姆定律同列器件可同时实现输入电压与电导的乘积操作. 同时利用基尔霍夫电压定律, RRAM 阵列能够在一个周期内完成矢量与矩阵的乘累加运算. 对应的数学表达式为

$$I_j = \sum_{i=1}^{n} V_i G_{ij}, \tag{1}$$

其中 V_i 为第 i 行的输入电压, G_{ij} 为存储器阵列中第 i 行、j 列的电导值, I_j 为第 j 列的输出电流值. 为了完善矩阵向量乘积方法, Pedretti 等[52] 探索了适配各类存内计算单元存储能力的均匀编码、位编码等矩阵表示方案, 如图 4(b) 所示, 与之对应的映射误差见图 4(c). 同时, 研究人员也探索了输入向量的信息编码方法, 提出了幅值、脉宽等时空编码方案等[52,53], 能够有效地提高数据传输密度和运算效率. 为了实现完备的正负输入和正负权重的运算, Park 等[54] 设计了如图 4(e) 的运算方法; Li 等[55] 探索了输出信息的模数转化方法. 然而, 在实际情况下, 向量矩阵乘积仍面临较多挑战. 例如, 通常存内计算器件并非理想的欧姆器件, 电压响应呈现非线性特性[56], 见图 4(f); 交叉阵列的线阻影响激励信号传输, 会引起运算偏差[57], 见图 4(g); 输出向量的模数转化方法实现方法仍不完善等.

4.1.2 向量-向量运算模式

如图 5(a) 所示, 基于交叉阵列的互连结构, 将行和列作为输入, 利用存内计算单元对外界施加的组合激励信号的响应作为输出结果, 直接存储在存

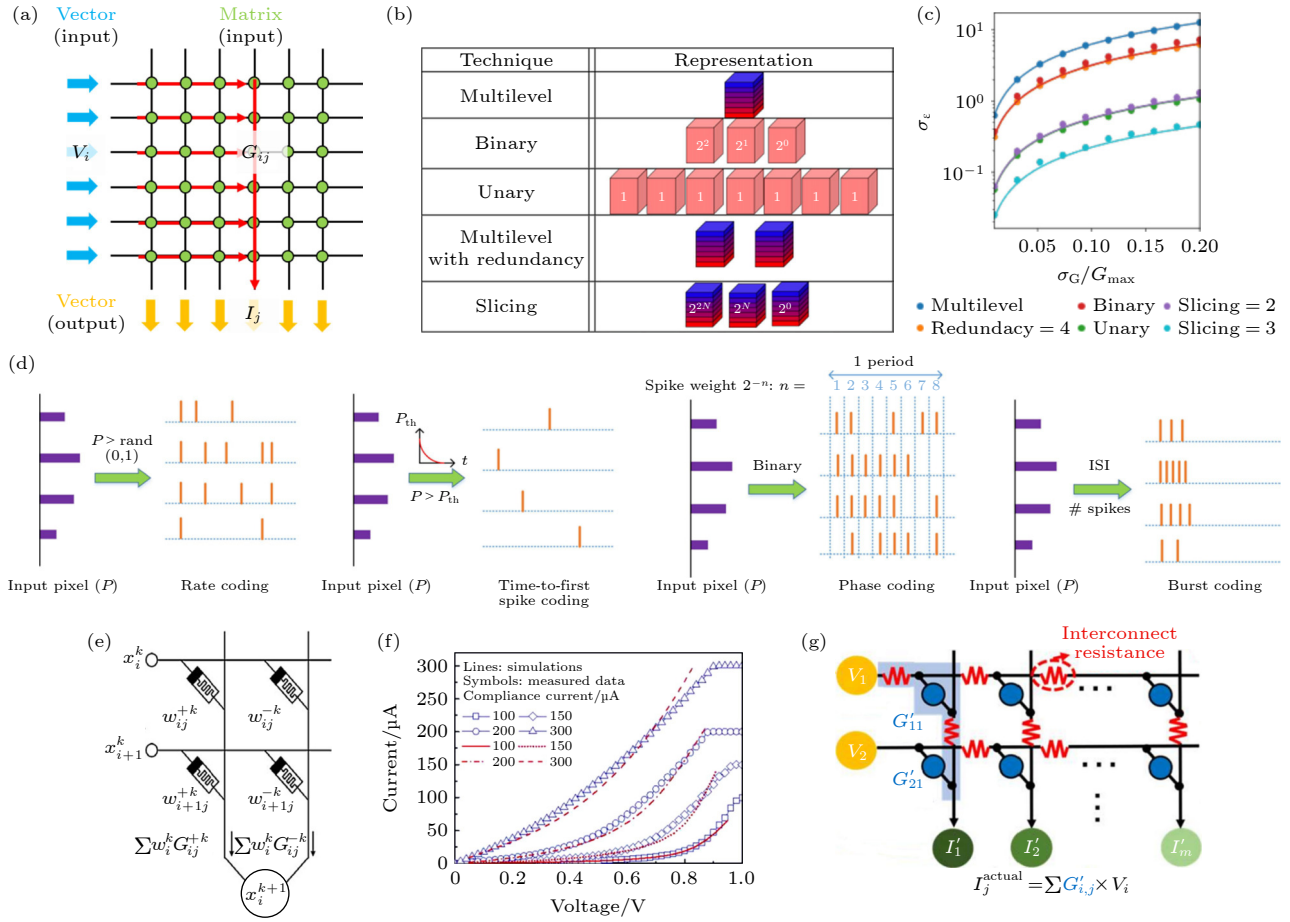

图 4 向量-矩阵运算模式 (a) 基本原理; (b) 矩阵编码模式[52]; (c) 器件状态波动性对编码的影响[52]; (d) 向量编码模式[53]; (e) 正负输入和权重的运算方法[54]; (f) 器件 I-V 非线性[56]; (g) 交叉阵列互联电阻[57]

Fig. 4. Vector-matrix operation mode: (a) The basic principle; (b) matrix encode (mapping) scheme[52]; (c) impact of device variation on the matrix encode[52]; (d) input vector encode schemes[53]; (e) the operation method of positive and negative input and weight[54]; (f) the nonlinearity I-V behavior of device[56]; (g) the interconnect resistance of cross-bar array[57].

内计算单元内, 从而实现了行向量与列向量的运算功能. 仍以两端器件 RRAM 为例, 利用两端信号的叠加作用, 阵列中每个单元均受其所在行和列的激励信号的影响. 具体数学表达式为

$$S_{ij}^{t+1} = f(V_i^1, V_j^2, S_{ij}^t), \tag{2}$$

其中, V_i^1 为第 i 行的电压, V_j^2 为第 j 列的电压, S_{ij}^t 为第 t 时刻位于第 i 行、j 列的 NVM 器件的存储状态, $f(\cdot)$ 为 NVM 器件在不同存储状态下, 对激励信号的影响关系. 利用这一特性 Liao 等[58] 提出了向量-向量乘积方案, 如图 5(b) 所示, 通过将运算矩阵拆解为向量的形式来实现完整的矩阵运算. 2019 年, Ambrogio 等[59] 实现了时间依赖可塑性学习规则, 如图 5(c) 所示, 用于类脑计算的学习法则; 另外, 可将器件的累积特性视为加法功能, 从而可以实现基于存内计算的半加器[60], 见图 5(d). 实际情况下, 向量-向量运算模式依然面临诸多问题, 如存储状态对单元激励响应的影响, 激励信号的编码方案、互联电阻的影响, 存内计算单元的存储容量限制 (图 5(e))[61] 等.

4.2 存内数字运算

计算机硬件基于二进制数据的表示和处理, 布尔逻辑在硬件体系结构中结构的描述、构建和优化过程方面扮演着十分重要的角色, 也是存内数字运算的研究目标. 常见的布尔逻辑如图 6(a) 所示, 其中 NAND, NOR 和蕴含逻辑 (implication, IMP) 是完备的逻辑形式, 仅利用任意一种逻辑, 通过组合就可实现其他所有逻辑类型, 成为存内数字运算硬件实现的首要目标. 如图 6(b), (c) 所示, 利用传统挥发型存储器 SRAM/动态随机存取存储器 (dynamic random access memory, DRAM) 均可实现完备的布尔逻辑功能. 然而, 面向大数据应用, 直

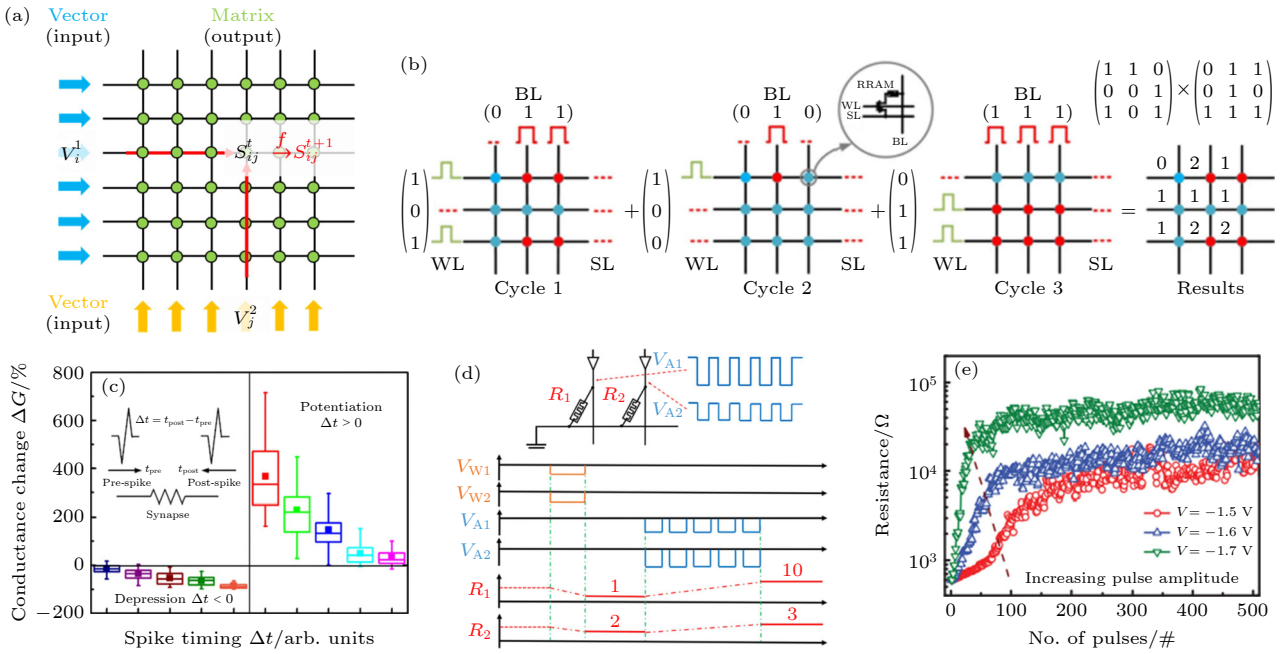

图 5 向量-向量运算模式 (a) 基本原理; (b) 向量形式的矩阵-矩阵乘积运算[58]; (c) 尖峰时间依赖可塑性学习规则[59]; (d) 一种基于 RRAM 的半加器实现方式[60]; (e) 典型 NVM 器件存储状态饱和限制[61]

Fig. 5. Vector-vector operation mode: (a) The basic principle; (b) the matrix-matrix multiplication based on vector form[58]; (c) the spike time dependent plasticity learning rule[59]; (d) the half-adder implementation approach based on RRAM[60]; (e) the saturation limited states range of typical NVM device[61].

接在非易失存储器内实现逻辑运算更具优势. 根据输入输出物理量的不同, 可以将存内数字运算分成 *V*-*R* 型、*R*-*V* 型、*V*-*V* 型和 *R*-*R* 型这 4 种逻辑形式. 以 RRAM 为例, 如图 6(d), *V*-*R* 型逻辑的输入 A 和 B 由施加在 RRAM 两端电极的电压高低表示, 逻辑结果为 RRAM 的存储状态. 2011 年, 亚琛工业大学[62] 通过利用双极型阻变器件实现了 *V*-*R* 型逻辑. 在此基础上, 华中科技大学[63] 利用 1T1R 单元的栅、源、漏和 RRAM 的阻态作为逻辑输入, 实现了广义上的 *V*-*R* 逻辑, 能够在两步操作内实现任意 16 种布尔逻辑. 由于 *V*-*R* 逻辑的输出结果为存内计算单元的存储状态, 具备原位存储特征, 是一种高效的逻辑实现方法. 然而, 由于输入输出变量统一, 级联需借助额外的信号转换电路. 如图 6(e) 所示, *R*-*V* 型逻辑将 RRAM 的存储状态作为逻辑输入[64], 通过对 A 和 B 同时施加读电压, 比较公共节点与基准值, 实现组合逻辑. 通过设置不同的基准值, 可以实现 NAND, AND, OR, NOR, XOR 和 XNOR 这 6 种常见逻辑. 与 *V*-*R*型逻辑类似, *R*-*V* 逻辑的输入输出物理量依然不统一, 也面临着无法直接级联的问题. *V*-*V* 型逻辑如图 6(f) 所示, 输入输出物理量均为电压信号[65]. 与 *R*-*V* 型逻辑的固定读电压不同, *V*-*V* 型逻辑通过是否施加读电压来代表输入 1 和 0, 在输出端同样使用基准值进行比较, 得到输出电压值[65]. 可见, *V*-*V* 型逻辑具备的直接级联的优势, 但其输出结果是易失的. *R*-*R* 型逻辑是利用 RRAM 件与辅助电阻的分压原理实现逻辑运算, 如图 6(g) 所示, 其输入输出物理量均为 RRAM 的存储状态[66]. *R*-*R* 型逻辑是最典型的非易失逻辑, 具备直接级联的优势. 同时, *R*-*R* 型逻辑的输入输出均为 RRAM 的存储状态, 因此具备非易失特性.

尽管存内数字运算研究进展显著, 但距离实际应用仍存在许多关键科学问题需要解决. 在器件层次, 存内计算单元的一致性和擦写次数有待提高, 以满足逻辑运算的准确性和频繁性; 在阵列层次, 泄漏电流和线阻等问题有待解决, 以推动存内数字运算向大规模电路的方向发展; 在逻辑级联层次, 单步逻辑的可靠性问题被进一步放大, 实现复杂逻辑运算面临较大的挑战.

4.3 存内计算模式的综合比较

表 1 对比了上述存内计算模式, 可以得出以下结论. 第一, 存内计算的计算模式呈多样化发展趋势, 这种多样化趋势归根于应用场景中运算的多样化需求. 由于利用存储器件实现各类运算功能存在

(a)

A	B	NOT A	A AND B	A OR B	A NOR B	A NAND B	A XOR B	A XNOR B	A IMP B	A NIMP B	TURE	A BUFFER
1	1	0	1	1	0	0	0	1	1	0	1	1
0	1	1	0	1	0	1	1	0	1	0	1	0
1	0	0	0	1	0	1	1	0	0	1	1	1
0	0	1	0	0	1	1	0	1	1	0	1	0

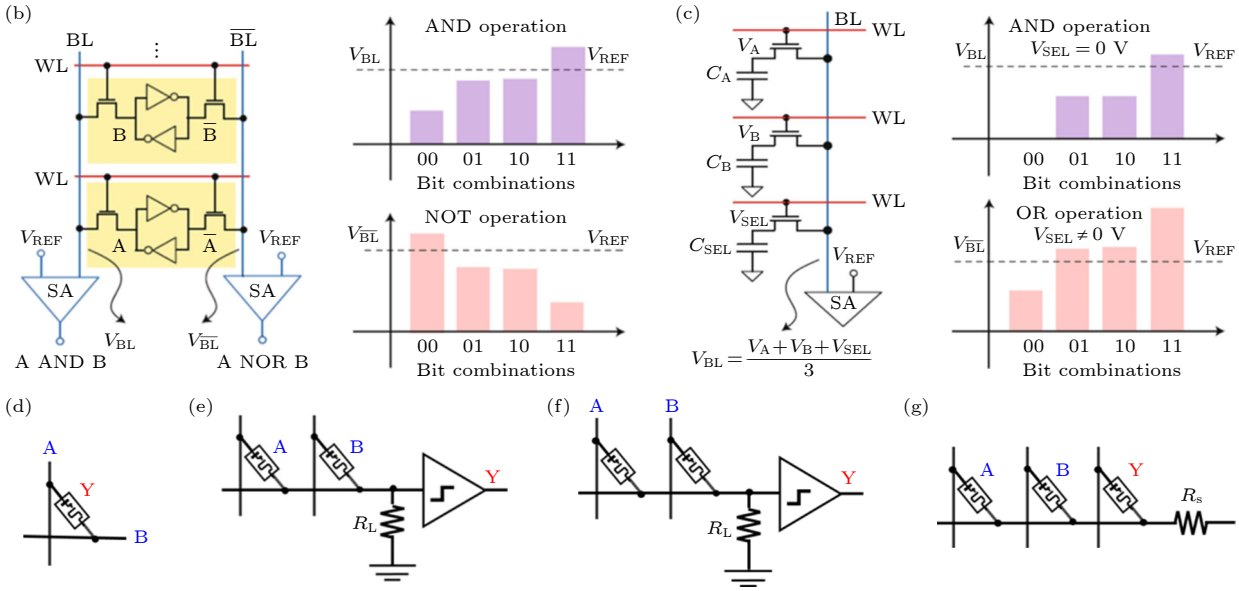

图 6 存内数字运算模式 (a) 常见逻辑真值表; 基于 SRAM (b) 和 DRAM (c) 的逻辑实现方案示例[26]; 基于 NVM 器件的逻辑 (d) *V*-*R* 型, (e) *R*-*V* 型, (f) *V*-*V* 型, (g) *R*-*R* 型

Fig. 6. In-memory digital computing mode: (a) The true value table of typical logic; SRAM (b) and DRAM (c) based logic implementation[26]; logics based on NVM device: (d) *V*-*R* type, (e) *R*-*V* type, (f) *V*-*V* type, (g) *R*-*R* type.

典型差异, 从而催生出不同类型的存内计算模式. 各类计算模式均在其特定的应用场景下发挥着重要作用. 第二, 存内模拟运算充分利用了存储器件的多值和模拟特性, 使单位面积的计算密度得到显著提升. 通过利用器件及其互联结构的电学特性, 存内模拟计算能够在模拟域进行诸如乘法、加法的代数逻辑运算, 从而突破布尔逻辑门的限制. 然而, 在功能器件的有限动态范围、编程精度等客观条件限制下, 存内模拟计算更加适合非精确计算的应用场景. 第三, 存内数字运算本质上是利用功能器件实现存算一体的完备布尔逻辑. 由于具备原位的计算和存储能力, 存内数字运算相较传统布尔逻辑, 具备更短距离的数据搬运特点和分布式计算的特征, 能够在数据密集型应用中承担数据预处理能力, 补充传统计算架构的不足. 然而, 受限于器件有限的鲁棒性和波动性, 在较为严苛的布尔逻辑运算的需求面前, 存内数字运算的可靠性有待进一步提升. 第四, 得益于器件的非易失特性、优秀的等比缩小能力和高并行度的阵列拓扑结构, 存内模拟运算和存内数字运算在各自擅长的领域内, 在能耗开销、运算效率和系统集成度等方面均具备一定的性能优势和发展潜力.

表 1 存内计算模式的特征

Table 1. Feature of in-memory computing modes.

	存内模拟计算	存内数字运算
功能	布尔逻辑, 代数运算	布尔逻辑
优势	高运算密度, 高并行度, 缓解数据搬运	精确计算, 高并行度, 缓解数据搬运
挑战	运算精度, 模数转化	器件鲁棒性、波动性
应用	深度学习、类脑计算等	逻辑电路、嵌入式存储

5 非挥发型存内计算系统应用

5.1 深度学习硬件加速器

深度学习是第三代人工智能算法的典型代表, 在模式识别和自然语言处理等领域具备显著的性能优势[67]. 反向传播算法是深度学习的理论基础[68], 其特点是基于随机梯度下降方法, 利用大量

的样本和标签信息, 通过损失函数不断对多层网络的权重矩阵进行逐步更新, 不断趋近理想权值. 通过不断加深网络层数和权重参数规模, 深度学习在高维特征变换、信息过滤等方面取得了显著进步. 由于深度学习算法包含海量的网络权重, 算法执行过程包含大量的特征图像与权重矩阵的乘积求和运算, 十分契合存内计算模拟运算, 被认为是存内计算的典型应用场景[69].

如图 7(a) 所示, 深度学习算法种类繁多, 囊括了全连接网络、卷积神经网络和循环神经网络等众多拓扑结构[70]. 其中, 全连接结构是深度学习的基本结构, 其前后神经元相互连接, 等价于交叉阵列结构构成的行列互连结构[50], 见图 7(b); 卷积神经网络是在全连接结构下, 进行结构化稀疏, 通过选通特定的行列也可在交叉阵列结构中实现; 循环神经网络则是引入时间因素, 将输出结果作为部分或全部信息, 再次输入到互联的网络内进行迭代计算, 同样将交叉结构作为基础. 如图 7(c) 所示, 深度学习硬件加速器根据功能不同, 可分为推理功能加速和训练功能加速. 实现推理功能时, 需将算法权重映射至存内计算单元内, 利用存内计算模拟运算的性能优势, 加速网络的前传计算能力; 实现训练功能时, 不仅需要推理功能实现前传和反传运算, 同时还要求存内计算单元能够实现原位的权重更新.

针对深度学习硬件加速研究, 研究人员分别在器件层次、运算核层次和系统层次展开了深入研究. 在器件层次, 深度学习硬件加速器面临着器件性能的有限表现与算法权重较高需求的不适配问题, 主要表现为器件有限的动态范围、编程精度、存储状态波动、信号响应非线性和非对称等非理想因素. 为克服这一系列问题, 研究人员提出了各类算法在不同运算精度下的权重映射方法. 2019 年, Huang等[71] 研究了利用非线性非对称器件进行权重更新的方法 (图 7(d)); 2021 年, Feng 等[72] 研究了器件的权重调制方法、优化了器件的状态保持特性 (图 7(e), (f)). 同时, 深度学习算法也朝着轻量化方向发展, 衍生出硬件更加友好的二值化、少值化的算法变体. 在运算核层次, 面临着利用存储阵列实现高效数据传输和模数混合运算的挑战, 不仅要克服阵列结构本身的串扰、线阻等问题, 也需要解决外围控制电路的设计实现问题. 相关的研究工作在数据搬运方法、后处理模数转化实现方案、无需模数转化的多级直连传输方法等方面取得了一定的研究进展[73,74]. 在系统层次, 如图 7(g) 所示, 需在保证运算效率的前提下解决有限硬件资源的分配问题[75]. 研究人员提出了图 7(h) 所示的流水线式网络映射方法, 研究了运算核的协同工作模式以及多核存内计算架构[76,77] 等问题. 同时, 根据训练和推理的不同功能特点, 需进行适当的加速器架构研究. 研究人员基于 NVM 器件提出了兼容多种算法的存内计算系统框架[78,79], 并验证了存内计算技术的性能优势. 面向未来, 深度学习的存内计算硬件加速研究依然面临着图 7(i) 所示的稀疏神经网络的适配性问题, 也需要解决图 7(j) 所示的输入信息的数据搬运问题[74].

5.2 类脑计算

当前的人工智能技术仍然存在一定局限性, 相比人脑的功能多样性和复杂度仍存在明显差距. 为了更接近人脑功能, 人们提出了神经形态计算的概念. 预期的神经形态计算系统, 具备在功能上模拟脑、性能上趋近脑、规模上超越脑的典型特征, 这被认为是未来人工智能的发展方向, 也将是存内计算技术的重要应用场景. 仿脑 (brain-like) 和类脑 (brain-inspired) 神经网络都属于神经形态硬件系统的研究范畴. 前者侧重模仿人脑神经网络的工作模式, 注重模拟生物神经元、突触等基本单元的功能, 以期望更接近人脑的工作模式; 后者偏向在生物的基础上抽象数学模型, 构建基本单元的数学模型并发展相应的算法理论. 仿脑为类脑提供了硬件基础, 类脑拓展了仿脑的发展空间, 二者相辅相成, 构成了当前神经形态硬件系统的发展方向.

如图 8(a)—(g) 所示, 在仿脑领域研究人员利用存内计算功能器件实现生物突触的长程可塑性 (long-time plasticity, LTP)、短程可塑性 (shot-time plasticity, STP) 和长短程可塑性的转变, 实现了生物突触的双脉冲易化特性[80], 实现了尖峰脉冲时间依赖和频率依赖可靠性[80–82] 等一系列突触基本功能; 如图 8(h), (i) 所示, Li 等[83] 利用存内计算功能器件实现了生物神经元的阈值特性、非线性信号调制能力和信号激励特性等. 在类脑领域, Lashkare 等[84] 利用存内计算功能单元实现了积分触发和泄漏积分触发神经元, 实现了基于尖峰时间和频率依赖可塑性的学习法则. Milo 等[85] 利用全连接网络演示了无监督学习能力 (图 8(k));

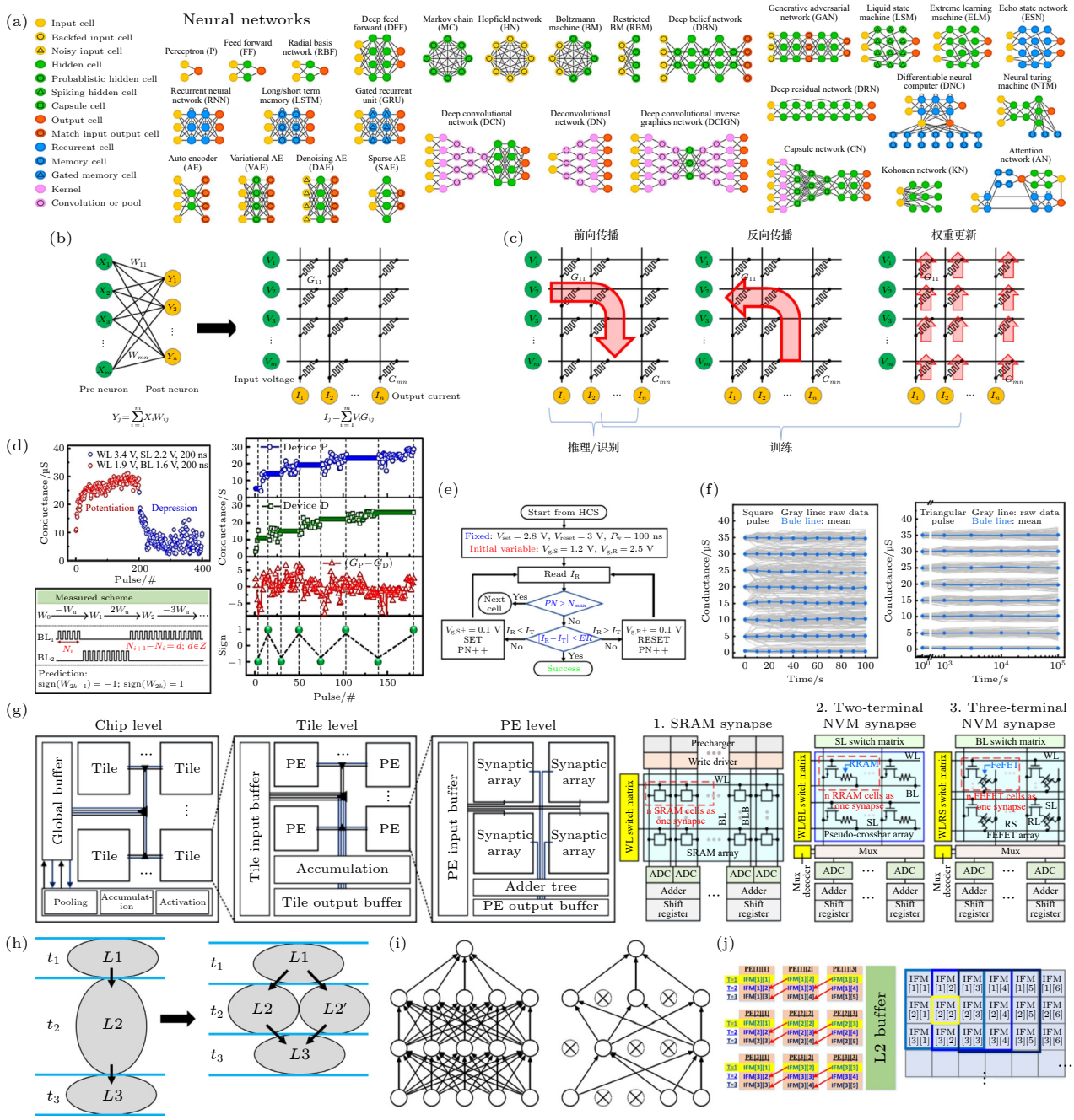

图 7 存内计算加速深度学习 (a) 常见深度学习算法分类[70]; (b) 存内计算加速深度学习的基本原理[50]; (c) 深度学习各功能的存内计算实现方式; (d) 利用二值神经网络算法克服器件非线性的影响[71]; (e) 器件操作优化方案[72]; (f) 利用激励信号波形抑制器件波动性[72]; (g) 基于存内计算的深度学习加速器的典型架构[73]; (h) 流水线硬件实现方法加速网络运算效率; (i) 神经网络稀疏性表现形式, 结构化和非结构化; (j) 减少输入信息搬运的数据调用方案[74]

Fig. 7. In-memory computing based deep learning accelerator: (a) The classes of deep learning algorithms[70]; (b) the basic principle of in-memory computing accelerates deep learning algorithm[50]; (c) in-memory computing implementation of deep learning functions; (d) solve the impact of device non-linearity switch behavior by binarized neural network[71]; (e) the optimized programming scheme of device[72]; (f) improve the device reliability by optimizing the stimulus signal[72]; (g) typical architecture of deep learning accelerators based on in-memory computing[73]; (h) pipeline weight mapping approach to speed up network computing efficiency; (i) the sparsening of neural network: structured and unstructured; (j) data call scheme to reduce input information handling[74].

图 8(l) 实现了霍普菲德网络和联想学习功能[86] 等. 同时, 图 8(m)为 Larkum[87] 探索得更复杂的仿生神经网络模型. 基于存内计算的脉冲神经网络也在不断推进[88,89], 如图 8(n) 所示的可直接级联的脉冲神经网络[75]. 然而, 当前的神经形态计算研究仍存在巨大挑战. 在仿脑领域, 仍面临着如何完整揭

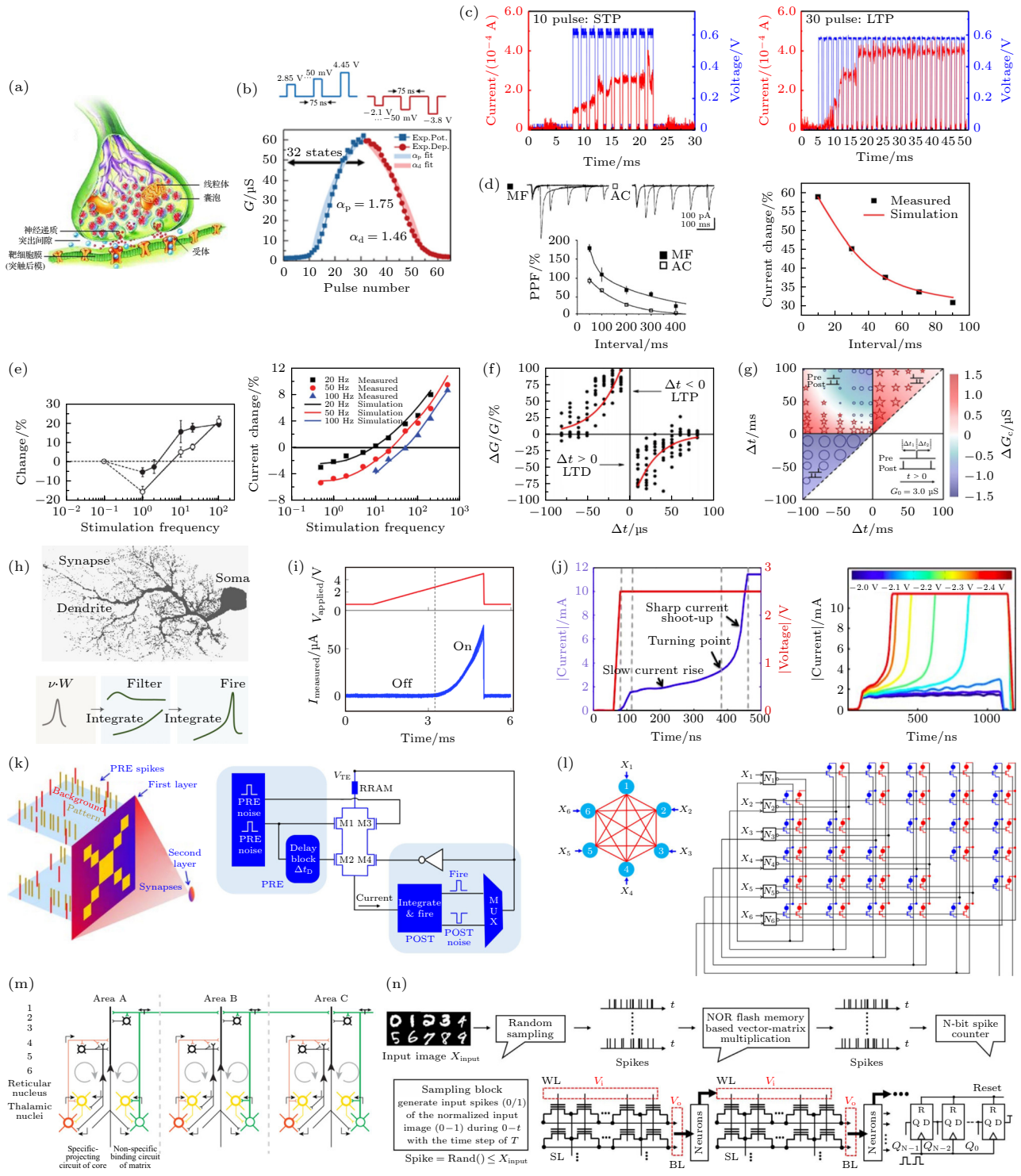

图 8 基于存内计算技术的类脑计算研究 (a) 生物突触结构; 利用 NVM 器件实现突触的 (b) LTP, (c) STP 和长短程可塑性转变; (d) 双脉冲易化响应特性[80]; (e) 尖峰脉冲频率依赖可塑性[80]; (f) 尖峰脉冲时间依赖可塑性[80]; (g) RRAM 中的 Bienenstock-Cooper-Munro 权重更新规则[82]; (h) 生物神经元结构[83]; (i) 基于 RRAM 的神经元树突非线性调制功能[83]; (j) 神经元积分触发功能[84]; (k) 基于尖峰脉冲频率依赖可塑性的脉冲神经网络非监督学习功能[85]; (l) 霍普菲德网络学习规则[86]; (m) 生物神经网络理论模型[87]; (n) 脉冲神经网络实现方案[75]

Fig. 8. Neuromorphic computing based on in-memory computing: (a) The biological synapse; (b) LTP, (c) STP and the conversation between STP and LTP of NVM device based artificial synapse; (d) double pulse facilitated response characteristics[80]; (e) the spike rate dependent plasticity (SRDP) [80]; (f) the spike-time dependent plasticity (STDP) [80]; (g) the Bienenstock-Cooper-Munro weight update rules in RRAM[82]; (h) the principle of biological neural[83]; (i) the signal modulation capability of the RRAM based artificial dendrite[83]; (j) neuron integration-fire function[84]; (k) unsupervised online training follows the spike rate dependent plasticity based spike neural network learning rule[85]; (l) the Hopfield eLearning rules[86]; (m) the model of biological neural network[87]; (n) implementation of spiking neural network[75].

示完备的人脑工作原理、如何在单一元件内集成多种仿生功能、如何在系统层次上融合各类生物功能等重要问题. 在类脑领域, 则面临着理论模型不完善, 网络算法功能单一、性能不足等显著问题, 仍然难以在结构和功能上模拟生物神经网络的完整功能. 二值化、少值化的脉冲神经网络是探索类脑计算系统功能的重要手段, 研究工作利用二值突触实现了低复杂度的硬件友好的脉冲神经网络[90].

5.3 非易失型布尔逻辑

2010 年, 惠普公司利用 RRAM 在特定电压脉冲下发生阻值变化的特点, 实现了一种状态逻辑计算功能[91]. 这种逻辑计算的基础是实质蕴含逻辑, 数学形式是“pIMPq” (图 9(a)). 结合 FALSE逻辑, 实质蕴含逻辑可以实现完整的二值数字逻辑函数, 从而形成了一种在存储器内实现完整逻辑计算的技术路线. 2014 年, 以色列理工学院利用 RRAM 的分压关系进行逻辑运算, 实现了 NOR 逻辑, 从而提出了一种名为 MAGIC 的 RRAM 辅助逻辑操作方案[92]. 2016 年, 北京大学 Huang 等[93] 通过改变 RRAM 上的电压信号施加方法实现了一种布尔代数逻辑, 即一步操作实现 NAND 和 AND 逻辑, 具备逻辑重构功能, 并建立了图 9(b) 所示的存内计算系统架构. 这三种实现方案各有优劣. 从结构上看, 实质蕴含逻辑和布尔代数逻辑均需负载电阻参与逻辑操作, 影响存内计算的集成密度. 从操作步骤看, 实质蕴含逻辑和 RRAM 辅助逻辑均只能实现一种基础逻辑操作, 故实现完整逻辑需较多操作步数, 复杂度较高. 从输入输出状态变化看, 实质蕴含逻辑的一个输入与输出共享同一个器件, 会改变输入信息. RRAM 辅助逻辑需要在满足特定条件的阻变器件中才能实现, 否则输入器件的存储状态也可能发生改变. 为了提升稳态逻辑的系统可靠性, 北京大学 Shen 等[94] 在 2019 年提出了基于

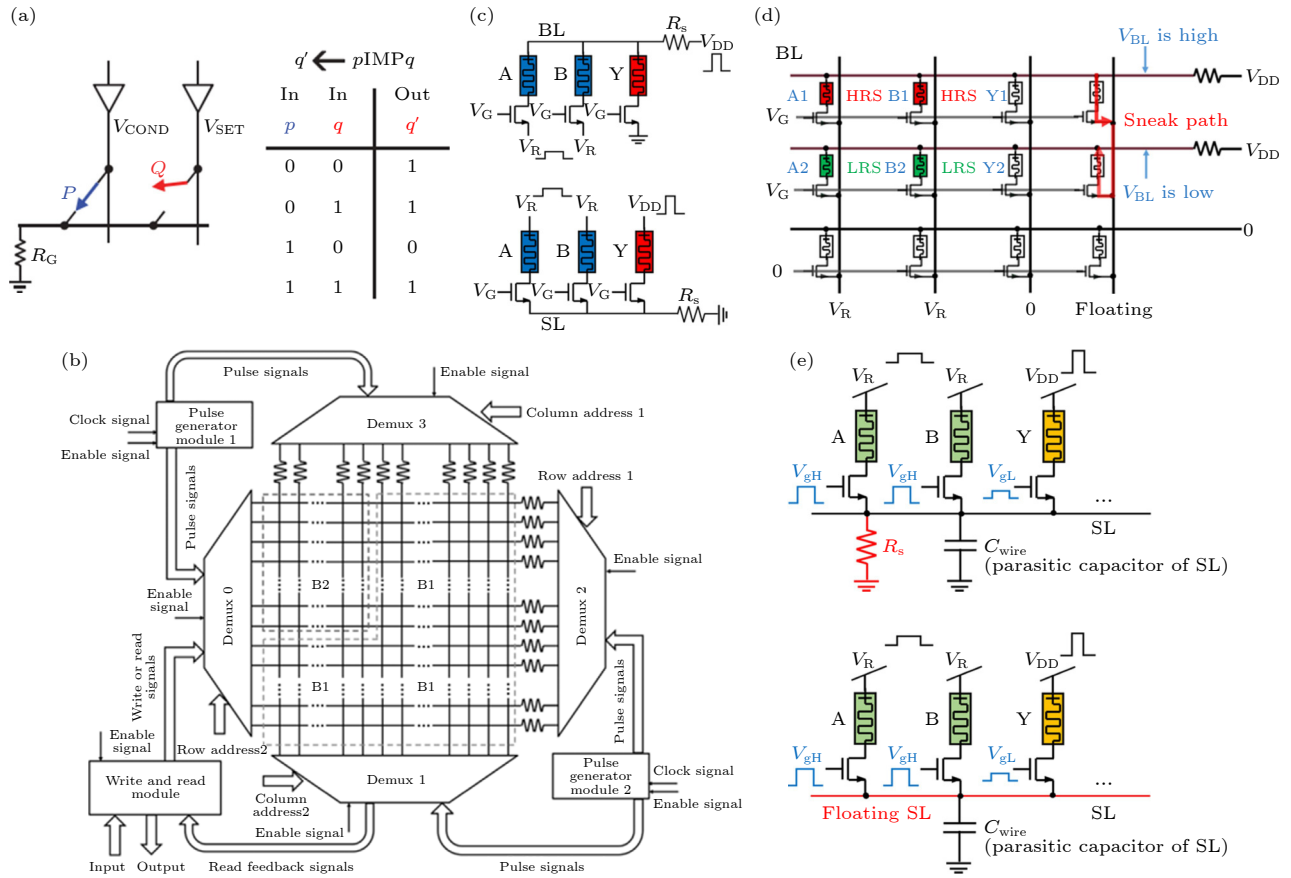

图 9 非易失状态逻辑 (a) IMP 逻辑实现方案[91]; (b) 状态逻辑运算核架构[93]; (c) 基于 1T1R 的状态逻辑[94]; (d) 利用 1T1R 结构抑制交叉阵列串扰[94]; (e) 利用寄生电容替代辅助 RRAM 的状态逻辑[95]

Fig. 9. Non-volatile stateful logic: (a) implementation scheme based IMP logic[91]; (b) the architecture of stateful logic process core[93]; (c) 1T1R based stateful logic[94]; (d) reduce the impact of sneak path by 1T1R structure[94]; (e) parasitic capacitor assisted RRAM based stateful logic[95].

1T1R 的状态逻辑实现方式, 如图 9(c) 所示, 用以抑制阵列中的泄漏电流. 2020 年, Shen 等 [95] 提出了一种基于寄生电容的状态逻辑实现方案, 从而省去了辅助电阻, 形成了更易于加工的阵列制备方案, 见图 9(e).

5.4 内容可寻址存储器

内容可寻址存储器 (content addressable memory, CAM) 是一种面向超高速数据搜索应用的一种存储系统, 如图 10(a) 所示. CAM 的工作原理与随机存取存储器相反, 输入为存储信息, 输出为存储地址. 其工作过程为将输入信息依次或并行与 CAM 内部信息比较, 输出匹配的存储地址. 根据 CAM 的单元存储状态, 分为二态内容可寻址存储器 (BCAM) 和三态内容可寻址存储器 (TCAM). TCAM 除了 0/1 外, 还包含“don't care”状态, 即该状态是否匹配不影响比较结果. 传统 CAM 通常使用 SRAM 作为基本存储单元, 见图 10(b). 然而, 尽管器件尺寸仍在不断缩小, SRAM 的面积开销和泄漏电流问题仍限制着 CAM 的进一步发展. 存内计算单元由于具备高密度存储能力、低读取功耗和 NVM 等特点, 是发展 CAM 技术的一种潜在方向. 2011 年, IBM 利用 PCM 器件演示了 CAM 和 TCAM 的基本功能 [96]. 相较传统基于 SRAM 的实现方式, 基于 PCM 的 CAM 在存储密度和能耗开销方面展现出超过 5 倍的性能优势. 匹兹堡大学 Yan 等 [97] 提出了基于 STT-MRAM 的 TCAM 模块, 并设计演示了 Dual-N 型和 P-N 型方案, 改善了搜索延时性能 (见图 10(c)). 2018 年, Grossi 等 [98] 研制出基于 RRAM 的 128 bit TCAM 宏, 搜索速度与传统 TCAM 齐平, 并取得了良好的系统可靠性表现 (图 10(d)). 图 10(e) 为北京大学 Yang 等 [99] 基于 3D NAND 型 flash 提出了一款超低功耗、高存储密度的 TCAM 实现方案. 结果如图 10(f) 所示, 通过比较读出电流与阈值电压, 可以分辨出搜索信息与存储信息的匹配与否. 预期每次搜索每比特能耗能够达到 0.298 fJ (64 bit word) 和大于 582 倍的存储密度 (96 layer).

5.5 线性方程组求解器

线性方程组是科学计算领域极其重要的运算需求, 广泛应用于天气预报、半导体器件仿真等实际场景. 在绝大多数情况下, 解析求解线性方程组都是不现实的. 通常利用数字求解的方法, 将解空间近似为离散网格, 通过高精度运算不断迭代, 得到满足实际精度需求的近似解. 矩阵-向量乘积是

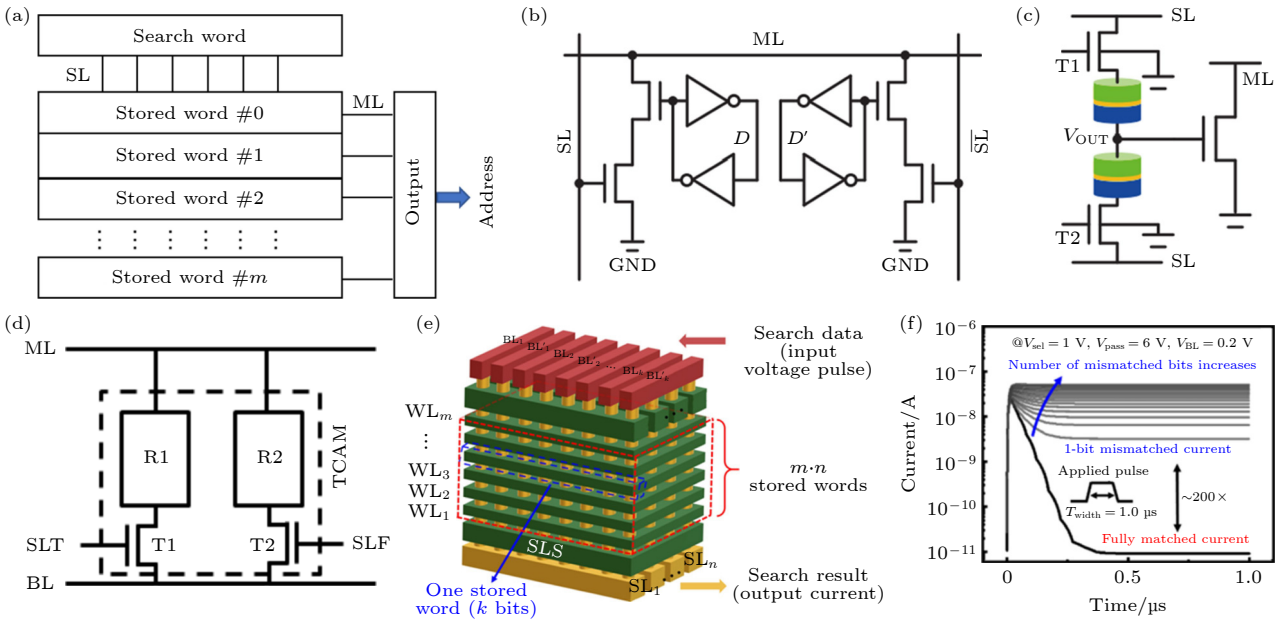

图 10 (a) CAM 基本实现方式 [91]; (b) 基于 SRAM 的 TCAM 基本单元 [93]; (c) 基于 STT-RAM 的 TCAM 基本单元 [97]; (d) 基于 RRAM 的 TCAM 基本单元 [98]; (e) 基于 NAND 型 flash 的 TCAM 实现方式 [99]; (f) 匹配与失配的输出结果示例 [99]

Fig. 10. (a) Typical structure of CAM[91]; (b) TCAM basic unit based on SRAM[93]; (c) TCAM basic unit based on STT-RAM[97]; (d) TCAM basic unit based on RRAM[98]; (e) based implementation of TCAM based on NAND flash[99]; (f) example of the matched and mismatched results[99].

迭代计算的关键步骤, 涉及大量的数据搬运和乘加运算. 存内模拟计算技术恰好在矩阵运算方面具备天然优势, 因此研究人员提出了基于存内计算技术的偏微分方程求解器. 2018 年, 如图 11(a) 所示, 密西根大学 Zidan 等[100] 提出了利用位切片技术的 16 bit 全 RRAM 偏微分方程求解器. 为了提升求解器性能, IBM 整合了高速低功耗的存内计算求解器和高精度的数字求解器的优势, 提出了混合精度架构, 齐平了 CPU、GPU 的处理能力[101], 见图 11(b). 2019 年, 米兰理工大学 Sun 等[19] 利用 RRAM 和负反馈阵列接连结构, 演示了一步求解线性方程, 如图 11(c)—(e) 所示. 之后, 这种技术方案又被扩展到求解薛定谔方程和其他代数问题[102]. 2021 年, 山东大学 Feng 等[103] 提出了存内计算技术加速浮点数尾数乘法的方法, 设计实现了 32 bit 浮点数求解器. 综上所述, 器件稳定性、一致性以及阵列的非理想因素等问题仍不可忽略, 依然影响着求解器的性能表现, 相应的算法和架构仍有较大改进空间.

5.6 其他应用场景

随机计算 (stochastic computing, SC) 是一种低成本的计算形式[104], 如图 12(a). 其工作原理是将信息量化为随机分布的 0/1 数据流, 利用数据流中的 0/1 比例表示实际信息, 由此可将乘法等运算简化为 AND 等基本逻辑操作, 相较传统计算具备较高的误差容忍度、运算逻辑简答等优势[104]. 存内计算功能器件是实现 SC 的重要技术路线[104,105]. 一方面, 存内计算功能器件是良好的随机信号发生器, 具备不同尺度下的噪声来源, 如图 12(b), (c); 另一方面, 存内计算功能在实现简单逻辑计算方面具备显著优势, 且具备高集成程度、高存储密度的特点.

物理不可克隆函数 (physically unclonable functions, PUF) 是一种利用某种物理内在机制构建的唯一性标识, 如图 12(d) 所示, 输入任意激励都会输出唯一且不可预测的响应, 常用作信息密钥、防伪等应用场景[106]. 存内计算功能器件, 特别是 RRAM 等新型 NVM 器件, 由于具备本质的不可预测的随机信号源, 是实现 PUF 的良好硬件基础[107,108]. 相关的系统架构设计如图 12(e) 所示, 其工作原理是利用 NVM 器件本征的不可预测、不可复制的器件波动性, 对给定外界激励信号形成不可克隆的、可重复触发的唯一响应[107].

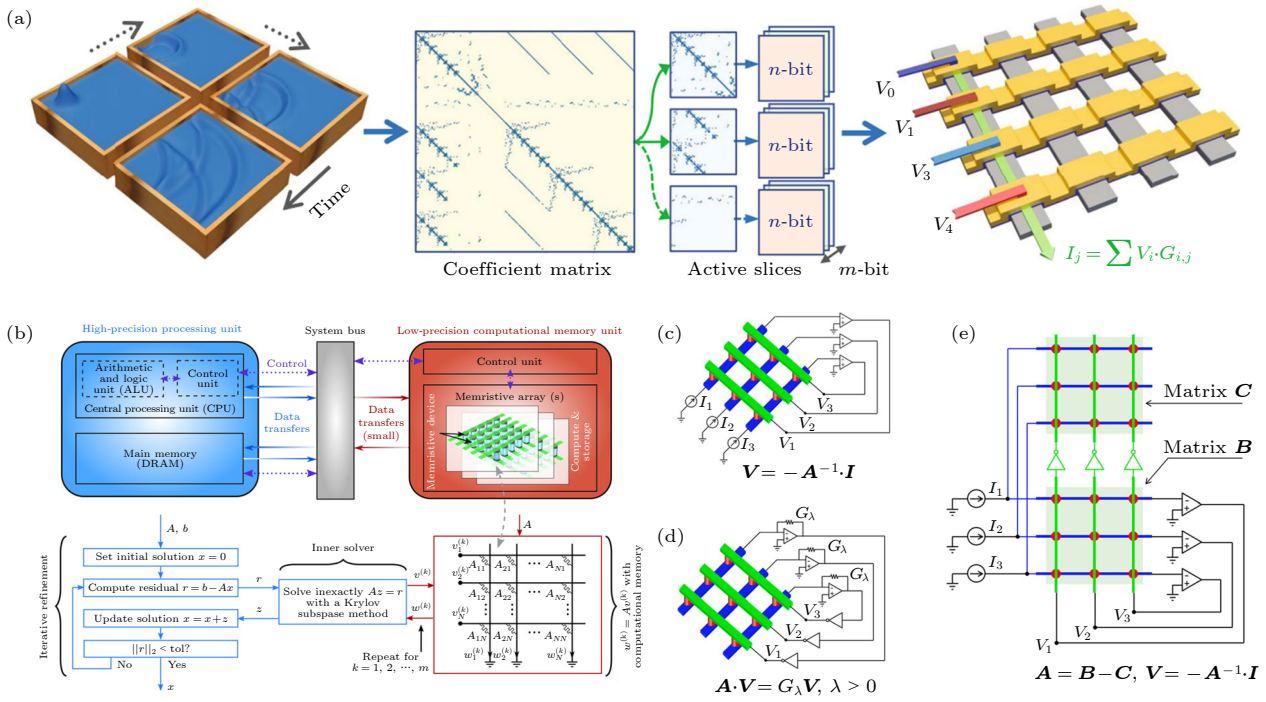

图 11 线性方程组求解器 (a) 高精度线性方程求解器实现方法[100]; (b) 混合精度求解器架构[101]; (c) 正权重矩阵求逆[19]; (d) 求解特征向量方程 $Ax = \lambda x$[19]; (e) 混合矩阵求逆[19]

Fig. 11. Linear equations solver: (a) The implementation of high-precision linear equation solver[100]; (b) the mixed-precision solver architecture[101]; (c) inverting a positive weight matrix[19]; (d) solve eigenvector equation $Ax = \lambda x$ [19]; (e) inverting a mixed matrix[19].

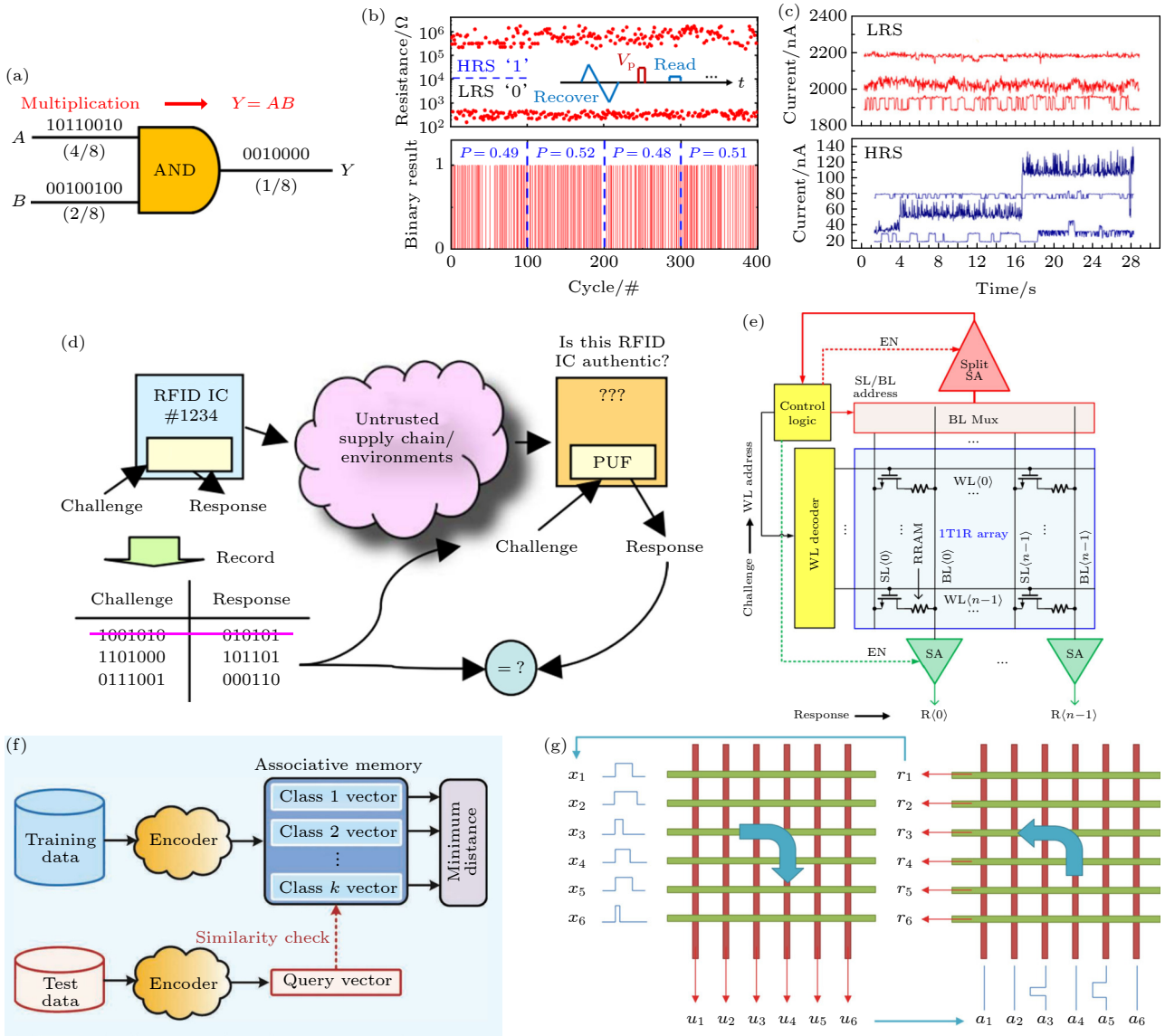

图 12 (a) SC 乘法工作原理[104]; (b) RRAM 阻变过程的随机性[104]; (c) 随机电报噪声特性; (d) 基于 PUF 的射频识别工作原理[106]; (e) 基于 RRAM 器件的 PUF 架构[107]; (f) HDC 分类原理[109]; (g) 基于 NVM 器件交叉阵列的稀疏编码[113]

Fig. 12. (a) The multiplication operation realized by SC[104]; (b) the random behavior of RRAM[104]; (c) noise characteristics of random telegram; (d) the operating principle of PUF based radio frequency identification[106]; (e) the architecture of RRAM based PUF[107]; (f) classification overview with HDC[109]; (g) sparse coding in NVM device based crossbar array[113].

此外, 超维计算 (hyperdimensional computing, HDC) 是一种新兴的计算方法, 如图 12(f) 所示, 通过在超维空间计算超维向量化的特征图像的相对距离, 实现分类识别功能[109]. 为方便理解, 可以将 HDC 的工作原理归纳为编码、搜索两个阶段. 在编码阶段, 利用算法将训练库数据的特征图像提取出来, 并编码为特征向量. 在搜索阶段, 将新产生的特征向量与存储器内部的特征向量进行对比, 剔除重复的冗余信息, 记忆具备显著区别的特征向量. HDC 可以使人工智能系统对过去感知的事情形成记忆, 以便更好地完成未来类似的任务. 存内计算由于具备高密度存储和运算功能, 能够同时实现 HDC 编码器和关联存储器, 在实现超维计算方面具有显著优势[110,111]. 利用存内计算功能器件和交叉阵列结构的矩阵结构, 如图 12(g) 所示, 研究人员开发出新型的图像压缩功能[112]、稀疏编码[113]等非易型存内计算技术的多种新型应用场景.

6 结 论

存内计算技术通过在基本单元上集成存储和运算功能, 能够显著减少数据搬运, 是突破传统

冯·诺依曼瓶颈和存储墙的一种新型计算范式. NVM 器件是实现存内计算的理想硬件载体, 不仅具备非挥发、低功耗等性能优势, 而且可以在器件层级实现存算功能融合, 从而构建高集成度、低功耗的存内计算硬件系统. 从工艺成熟的 flash 器件到潜力巨大的 RRAM 器件, 一脉相承的非挥发型存内计算技术体系, 在短期内具备成熟硬件开发能力, 在长期将拥有广阔的应用拓展空间, 是未来计算技术的重要发展方向. 基于 NVM 器件的存内计算各类实现方式均拥有各自的性能优势与适用场景, 为未来计算形态的发展提供了多种可能性. 总体来看, 存内模拟计算充分开发了 NVM器件的性能潜力, 在非精准运算场景下具备显著的低功耗、高集成度的性能优势. 其中向量-矩阵模式适用于以深度学习为典型代表的数据流驱动的运算任务; 向量-向量模式则更加契合类脑计算中前馈、反馈等多激励耦合的应用场景. 相对而言, 存内数字计算的高可靠性和存算融合能力, 更加兼容当前主流的计算平台, 能够作为协处理器来增强传统计算平台处理数据密集型任务的能力. 当然, 当前的非挥发型存内计算技术仍然面临诸多挑战.

1) 存内计算运算宏. 随着非挥发存内计算功能器件的性能逐步稳定、运算模式的设计日趋完善, 构建存内计算运算宏成为当前重要的研究课题. 存内计算运算宏涉及功能定义、电路设计等方面, 既需克服模数转化问题、工艺兼容性问题等硬件相关科学问题, 也需进行运算任务分解方法、数据分配机制等配套算法研究.

2) 系统架构设计. 存内计算系统在硬件上拥有存算融合的典型特征, 在处理任务的过程中, 需在大量存储数据中进行特定的运算操作, 其数据存储格式、中间数据搬运过程等与传统计算系统存在显著差异. 因此, 该系统架构的研发面临多方面的创新需求, 包括算法映射方法、多核协作机制等关键技术, 以及与传统计算体系互动互通的实现方法.

3) 硬件规模扩展与应用落地. 尽管非挥发型存内计算技术在深度学习、线性方程求解等诸多应用上展现出显著的性能优势. 然而, 受限当前存内计算系统有限的硬件规模, 其计算能力、系统功能仍十分有限, 难以承接实际的应用任务, 也就无法与主流计算平台进行技术竞争. 扩大非挥发型存内计算系统硬件规模、推动应用落地, 将是促进存内计算技术良性发展的重要途径.

参考文献

[1] Shah S H, Yaqoob I 2016 *IEEE SEGE* **1** 381
[2] Reinsel D, Gantz J, Rydning J 2017 *Don't Focus on Big Data* **1** 2
[3] Waldrop M M 2016 *Nat. News* **530** 144
[4] Backus J W 1978 *Comm. ACM.* **21** 613
[5] McKee S A 2004 *Proceedings of the 1st Conference on Computing frontiers* Ischia Italy, April 14–16, 2004 p162
[6] Indiveri G, Liu S C 2015 *Proceedings of the IEEE* **103** 1379
[7] Yang J J, Strukov D B, Stewart D R 2013 *Nature Nanotech.* **8** 13
[8] Chen B, Cai F, Zhou J, Ma W, Sheridan P, Lu W D 2015 *IEEE IEDM* Washington, December 7–9, 2015 p17.5.1
[9] Zahoor F, Zulkifli T Z A, Khanday F A 2020 *Nanoscale Res. Lett.* **15** 1
[10] Ma Y, Du Y, Du L, Lin J, Wang Z 2020 *GLSVLSI'20* Virtual Event, China, September 7–9, 2020 p265
[11] Xu X, Luo Q, Gong T, Lv H, Long S, Liu Q, Chuang S S, Li J, Liu M 2016 *IEEE VLSI* Honolulu, April 25–27, 2016 p1
[12] Chen A 2016 *Solid State Electron.* **125** 25
[13] Tao L, Xu R, Tian T, Xiang Z, Li Y, Jin X, Ren J, Li Z, Li C 2019 *MobiSys'19* Seoul, June 17–21, 2019 p612
[14] Lee J, Park B G, Kim Y 2019 *IEEE EDL.* **40** 1358
[15] Freitas R F, Wilcke W W 2008 *IBM J. Res. Dev.* **52** 439
[16] Wu W, Wu H, Gao B, Yao P, Zhang X, Peng X, Yu S, Qian H 2018 *IEEE VLSI* Honolulu, June 18–22, 2018 p103
[17] Wang Z, Wu H, Burr G W, Hwang C S, Wang K L, Xia Q, Yang J J 2020 *Nat. Rev. Mater.* **5** 173
[18] Xiang Y, Huang P, Zhao Y, Zhao M, Gao B, Wu H, Qian H, Liu X, Kang J F 2019 *IEEE Trans. Electron. Devices* **66** 4517
[19] Sun Z, Pedretti G, Ambrosi E, Bricalli A, Wang W, Ielmini D 2019 *Pro. Nat. Acad. Sci.* **116** 4123
[20] Zanotti T, Puglisi F M, Pavan P 2020 *IEEE Trans. Device Mater. Reliab.* **20** 278
[21] Yan B, Li B, Qiao X, Xue C X, Chang M F, Chen Y, Li H 2019 *Adv. Intell. Syst.* **1** 1900068
[22] Schuiki F, Schaffner M, Gürkaynak F K, Benini L 2018 *IEEE Trans. Comput.* **68** 484
[23] McClanahan C 2010 *A Survey Paper* **9** 1
[24] Wang Y E, Wei G Y, Brooks D 2020 *Pro. Mach. Learning Syst.* **2** 30
[25] Liu S, Du Z, Tao J, Han D, Luo T, Xie Y, Chen Y, Chen T 2016 *ACM/IEEE ISCA* Seoul, June 18–22, 2016 p393
[26] Sebastian A, Le Gallo M, Khaddam-Aljameh R, Eleftheriou E 2020 *Nature Nanotech.* **15** 529
[27] Si X, Chen J J, Tu Y N, Huang W H, Wang J H, Chiu Y C, Wei W C, Wu S Y, Sun X, Liu R, Yu S 2019 *IEEE J Solid-State Circuits* **55** 189
[28] Zhou Z, Huang P, Xiang Y C, Shen W S, Feng Y L, Gao B, Wu H Q, Qian H, Liu L F, Zhang X, Liu X Y, Kang J F 2018 *IEEE IEDM* San Francisco, November 29–December 07, 2018 p20.7.1
[29] Jerry M, Chen P Y, Zhang J, Sharma P, Ni K, Yu S, Datta S 2017 *IEEE IEDM* San Francisco, December 4–6, 2017 p6.2.1
[30] Peng X, Chakraborty W, Kaul A, Shim W, Bakir M S, Datta S, Yu S 2020 *IEEE IEDM* San Francisco, December 10–18, 2020 p30.4.1
[31] Xiang Y C, Huang P, Zhou Z, Han R Z, Jiang Y N, Shu Q M, Su Z Q, Liu Y B, Liu X Y, Kang J F 2019 *IEEE ISCAS* Sapporo Convention Center, May 26–29, 2019 p1

[32] Jiang H, Huang S, Peng X, Yu S 2020 *IEEE ISCAS* Spain, May 17–20, 2020 p1

[33] Khwa W S, Chen J J, Li J F, Si X, Yang E Y, Sun X, Liu R, Chen P Y, Li Q, Yu S, Chang M F 2018 *IEEE ISSCC* San Francisco, February 4–8, 2018 p496

[34] Guo R, Liu Y, Zheng S, Wu S Y, Ouyang P, Khwa W S, Chen X, Chen J J, Li X, Liu L, Chang M F, Wei S, Yin S 2019 *IEEE VLSI* Kyoto, June 9–14, 2019 p120

[35] Wang P, Xu F, Wang B, Gao B, Wu H, Qian H, Yu S 2018 *IEEE TVLSI* **27** 988

[36] Bez R, Camerlenghi E, Modelli A, Visconti A 2003 *Pro. IEEE* **91** 489

[37] Bayat F M, Guo X, Klachko M, Do N, Likharev K, Strukov D 2016 *74th Annual Device Research Conference (DRC)* Newark, June 19–22, 2016 p1

[38] Guo X, Bayat F M, Bavandpour M, Klachko M, Mahmoodi M R, Prezioso M, Likjarev K K, Strukov D B 2017 *IEEE IEDM* San Francisco, December 4–6, 2017 p6.5.1

[39] Han R, Huang P, Xiang Y, Liu C, Dong Z, Su Z, Liu Y, Liu L, Liu X, Kang J F 2019 *IEEE TCAS-I* **66** 1692

[40] Kim M, Liu M, Everson L, Park G, Jeon Y, Kim S, Lee S, Song S, Kim C H 2019 *IEEE IEDM* San Francisco, December 9–11, 2019 p38.3.1

[41] Lue H T, Hsu P K, Wei M L, Yeh T H, Du P Y, Chen W C, Wang K C, Lu C Y 2019 *IEEE IEDM* San Francisco, December 9–11, 2019 p38.1.1

[42] Tyagi V V, Buddhi D 2007 *Renewable and sustainable energy reviews* **11** 1146

[43] Khvalkovskiy A V, Apalkov D, Watts S, Chepulskii R, Beach R S, Ong A, Tang X, Driskill-Smith A, Butler W H, Visscher P B 2013 *J. Phys. D* **46** 074001

[44] Mikolajick T, Dehm C, Hartner W, Kasko I, Kastner M J, Nagel N, Moert M, Mazure C 2001 *Microelectron. Reliab.* **41** 947

[45] Wong H S P, Lee H Y, Yu S, Chen Y S, Wu Y, Chen P S, Lee B, Chen F, Tsai M J 2012 *Pro. IEEE* **100** 1951

[46] Huang P, Liu X Y, Chen B, Li H T, Wang Y J, Deng Y X, Wei K L, Zeng L, Gao B, Du G, Zhang X, Kang J F 2013 *IEEE TED* **60** 4090

[47] Baek I G, Lee M S, Seo S, Lee M J, Seo D H, Suh D S, Park J C, Park S O, Kim H S, Yoo I K, Chuang U I, Moon J T 2004 *IEEE IEDM* San Francisco, December 13 –15, 2004 p587

[48] Wang H, Yan X 2019 *Phys. Status Solidi (RRL)* **13** 1900073

[49] Rehman M M, Rehman H M M U, Gul J Z, Kim W Y, Karimov K S, Ahmed N 2020 *Sci. Technol. Adv. Mat.* **21** 147

[50] Zhang Y, Huang P, Gao B, Kang J K, Wu H Q 2020 *J. Phys. D* **54** 083002

[51] Shi T, Wang R, Wu Z, Sun Y, An J, Liu Q 2021 *Small Structures* **2** 2000109

[52] Pedretti G, Ambrosi E, Ielmini D 2021 *IEEE IRPS* Monterey, March 21–25, 2021 p1

[53] Guo W, Fouda M E, Eltawil A M, Salama K N 2021 *Front. Neurosci.* **15** 212

[54] Park J, Kwak M, Moon K, Woo J, Lee D, Hwang H 2016 *IEEE EDL* **37** 1559

[55] Li W, Sun X, Jiang H, Huang S, Yu S 2021 *IEEE ESSCIRC* Grenoble, September 13–22, 2021 p79

[56] Sung C, Padovani A, Beltrando B, Lee D, Kwak M, Lim S, Larcher L, Marca V D, Hwang H 2019 *IEEE J Electron Devi.* **7** 404

[57] Han L X, Xiang Y C, Huang P, Yu G H, Han R Z, Liu X Y, Kang J F 2021 *IEEE IRPS* Monterey, March 21–25, 2021 p1

[58] Liao Y, Wu H, Wan W, Zhang W, Gao B, Wong H S P, Qian H 2018 *IEEE VLSI* Honolulu, June 18–22, 2018 p31

[59] Ambrogio S, Balatti S, Milo V, Carboni R, Wang Z, Calderoni A, Ramaswamy N, Ielmini D 2016 *IEEE VLSI* Honolulu, June 14–16, 2016 p1

[60] Ma W, Zhou Z, Zhu D, Liu L 2016 *Electron. Lett.* **52** 1073

[61] Chen Z, Gao B, Zhou Z, Huang P, Li H, Ma W, Zhu D, Liu L, Liu X, Kang J F 2015 *IEEE IEDM* Washington, December 7–9, 2015, p17.7.1

[62] Rosezin R, Linn E, Kugeler C, Bruchhaus R, Waser R 2011 *IEEE EDL* **32** 710

[63] Gao S, Zeng F, Wang M, Wang G, Song C, Pan F 2015 *Sci. Rep.* **5** 15467

[64] Xie L, Du Nguyen H A, Yu J, Kaichouhi A, Taouil M, AlFailakawi M, Hamdioui S 2017 *IEEE ISVLSI* Bochum, July 3–5, 2017 p176

[65] Gao L, Alibart F, Strukov D B 2013 *IEEE T Nanotechnol.* **12** 115

[66] Li H, Gao B, Chen Z, Zhao Y, Huang P, Ye H, Liu L, Liu X, Kang J F 2015 *Sci. Rep.* **5** 1

[67] LeCun Y, Bengio Y, Hinton G 2015 *Nature* **521** 436

[68] Van Ooyen A, Nienhuis B 1992 *Neural Networks* **5** 465

[69] Tsai H, Ambrogio S, Narayanan P, Shelby R M, Burr G W 2018 *J. Phys. D* **51** 283001

[70] Leijnen S, Veen F 2020 *Multidisciplinary Digital Publishing Institute Proceedings* **47** 9

[71] Huang P, Zhou Z, Zhang Y, Xiang Y, Han R, Liu L, Liu X, Kang J 2019 *APL Mater.* **7** 081105

[72] Feng Y, Huang P, Zhao Y, Shan Y, Zhang Y, Zhou Z, Liu L, Liu X, Kang J F 2021 *IEEE EDL* **42** 1168

[73] Peng X, Liu R, Yu S 2019 *IEEE ISCAS* Sapporo Convention Center, May 26–29, 2019 p1

[74] Xiang Y, Huang P, Han R, Li C, Wang K, Liu X, Kang J F 2020 *IEEE TED* **67** 2329

[75] Peng X, Huang S, Luo Y, Sun X, Yu S 2019 *IEEE IEDM* San Francisco, December 9–11, 2019 p32.5.1

[76] Prabhu N L, Raghavan N 2021 *IEEE Access* **9** 168093

[77] Hsu T H, Lue H T, Hsu P K, Yeh T H, Du P Y, Lee G R, Chu C J, Wang K C, Liu C Y 2020 *IEEE IEDM*, San Francisco, December 10–18, 2020 p6.3.1

[78] Chi P, Li S, Xu C, Zhang T, Zhao J, Liu Y, Wang Y, Xie Y 2016 *ACM SIGARCH Computer Architecture News* **44** 27

[79] Cheng M, Xia L, Zhu Z, Cai Y, Xie Y, Wang Y, Yang H 2017 *ACM/EDAC/IEEE DAC* Austin, June 19–23, 2017 p1

[80] Du C, Ma W, Chang T, Sheridan P, Lu W D 2015 *Adv. Funct. Mater.* **25** 4290

[81] Huang P, Li Z, Dong Z, Han R, Zhou Z, Zhu D, Liu L, Liu X, Kang J F 2019 *ACS Appl. Electronic Mater.* **1** 845

[82] Wang Z, Zeng T, Ren Y, Lin Y, Xu H, Zhao X, Liu Y, Ielmini D 2020 *Nature Comm.* **11** 1

[83] Li X, Tang J, Zhang Q, Gao B, Yang J J, Song S, Wu W, Zhang W, Yao P, Deng N, Xie Y, Qian H, Wu H 2020 *Nature Nanotech.* **15** 776

[84] Lashkare S, Chouhan S, Chavan T, Bhat A, Kumbhaew P, Ganguly U 2018 *IEEE EDL* **39** 484

[85] Milo V, Pedretti G, Carboni R, Calderoni A, Ramaswamy N, Ambrogio S, Ielmini D 2016 *IEEE IEDM* San Francisco, December 3–7, 2016 p16.8.1

[86] Milo V, Ielmini D, Chicca E 2017 *IEEE IEDM* San Francisco, December 04–06, 2017 p11.2.1

[87] Larkum M 2013 *Trends Neurosci.* **36** 141

[88] Zhou Z, Liu C, Shen W, Dong Z, Chen Z, Huang P, Liu L,

Liu X, Kang J F 2017 *Nanoscale Res. Lett.* **12** 1

[89] Majdabadi M M, Shamsi J, Shokouhi S B 2021 *Analog Integr. Circ. S* **107** 249

[90] Tang H, Kim H, Kim H, Park J 2019 *IEEE T. Biomed. Circ. S* **13** 1664

[91] Borghetti J, Snider G S, Kuekes P J, Yang J J, Stewaert D R, Williams R S 2010 *Nature* **464** 873

[92] Talati N, Gupta S, Mane P, Kvatinsky S 2016 *IEEE T. Nanotechnol.* **15** 635

[93] Huang P, Kang J, Zhao Y, Chen S, Han R, Zhou Z, Chen Z, Ma W, Li M, Liu L, Liu X 2016 *Adv. Mater.* **28** 9758

[94] Shen W, Huang P, Fan M, Han R, Zhou Z, Gao B, Wu H, Qian H, Liu L, Liu X, Zhang X, Kang J F 2019 *IEEE EDL* **40** 1538

[95] Shen W, Huang P, Wang X, Feng Y, Xu W, Gao B, Wu H, Qian H, Liu L, Zhang X, Kang J F 2020 *IEEE EDTM* Penang, April 6–21, 2020 p1

[96] Rajendran B, Cheek R W, Lastras L A, Franceschini M M, Breitwisch M J, Schrott A G, Li J, Montoye R K, Chang L, Lam C 2011 *IEEE IMW* Monterey, May 22–25, 2011 p1

[97] Yan B, Li Z, Chen Y, Hai L 2016 *NVMTS* Pittsburgh, November 17–19, 2016 p1

[98] Grossi A, Vianello E, Zambelli C, Royer P, Noel J P, Giraud B, Perniola L, Olivo P, Nowak E 2018 *IEEE T. VLSI Syst.* **26** 2599

[99] Yang H Z, Huang P, Han R Z, Xiang Y C, Feng Y, Gao B, Chen J Z, Liu L F, Liu X Y, Kang J F 2020 *IEEE SNW* Honolulu, June 13–14, 2020 p29

[100] Zidan M A, Jeong Y J, Lee J H, Chen B, Huang S, Kushner M J, Lu W D 2018 *Nat. Electron.* **1** 411

[101] Gallo M L, Sebastian A, Mathis R, Manica M, Giefers H, Tuma T, Bekas C, Curioni A, Eleftheriou E 2018 *Nat. Electron.* **1** 246

[102] Sun Z, Ambrosi E, Pedretti G, Bricalli A, Ielmini D 2020 *IEEE TED* **67** 1466

[103] Feng Y, Chen B, Liu J, Sun Z H, Hu H Y, Zhang J Y, Zhan X P, Chen J C 2021 *IEEE IEDM* San Francisco, December 13–15, 2021 p12.1.1

[104] Shen W, Huang P, Fan M, Zhao Y, Feng Y, Liu L, Liu X, Kang J F 2020 *IEEE TED* **68** 103

[105] Suri M, Querlioz D, Bichler O, Palma G, Vianello E, Vuillaume D, Gamrat C, DeSalvo B 2013 *IEEE TED* **60** 2402

[106] Devadas S, Suh E, Paral S, Tom Z, Vivek K 2008 *IEEE International Conference On RFID* Las Vegas, April 16–17, 2018 p58

[107] Liu R, Wu H, Pang Y, Qian H, Yu S 2016 *IEEE HOST* McLean, May 3–5, 2016 p13

[108] Mahmoodi M R, Nili H, Strukov D B 2018 *IEEE VLSI* Honolulu, June 18–22, 2018 p99

[109] Ge L, Parhi K K 2020 *IEEE Circ. Syst. Mag.* **20** 30

[110] Karunaratne G, Le Gallo M, Cherubini G, Cherubini G, Benini L, Rahimi A, Sebastian A 2020 *Nat. Electron.* **3** 327

[111] Liu J, Ma M, Zhu Z, Wang Y, Yang H 2019 *IEEE ICECS* Genoa, November 27–29, 2019 p4

[112] Xu J, Feng D, Hua Y, Tong W, Liu J, Li C, Zhou W 2017 *IEEE ICCD* Boston Area, November 5–8, 2017 p573

[113] Sheridan P M, Cai F, Du C, Ma W, Zhang Z, Lu W D 2017 *Nat. Nanotechnol.* **12** 784

SPECIAL TOPIC—Physical electronics for brain-inspired computing

Non-volatile memory based in-memory computing technology

Zhou Zheng Huang Peng Kang Jin-Feng †

(*School of Integrated Circuits*, *Peking University*, *Beijing* 100871, *China*)

(Received 5 March 2022; revised manuscript received 10 June 2022)

Abstract

By integrating the storage and computing functions on the fundamental elements, computing in-memory (CIM) technology is widely considered as a novel computational paradigm that can break the bottleneck of Von Neumann architecture. Nonvolatile memory device is an appropriate hardware implementation approach of CIM, which possess significantly advantages, such as excellent scalability, low consumption, and versatility. In this paper, first we introduce the basic concept of CIM, including the technical background and technical characteristics. Then, we review the traditional and novel nonvolatile memory devices, flash and resistive random access memory (RRAM), used in non-volatile based computing in-memory (nvCIM) system. After that, we explain the operation modes of nvCIM: in-memory analog computing and in-memory digital computing. In addition, the applications of nvCIM are also discussed, including deep learning accelerator, neuromorphic computing, and stateful logic. Finally, we summarize the current research advances in nvCIM and provide an outlook on possible research directions in the future.

Keywords: in-memory computing, non-volatile memory, flash, resistive random access memory

PACS: 85.35.–p, 07.05.Mh, 84.35.+i, 85.30.Tv **DOI:** 10.7498/aps.71.20220397

† Corresponding author. E-mail: kangjf@pku.edu.cn