第 11 卷 第 3 期
2022 年 5 月
集 成 技 术
JOURNAL OF INTEGRATION TECHNOLOGY
Vol. 11 No. 3
May 2022

引文格式：
杜亚娟, 金凯伦, 王子烨, 等. 非易失性存储器件的性能、可靠性及应用 [J]. 集成技术, 2022, 11(3): 42-55.
Du YJ, Jin KL, Wang ZY, et al. Performance, reliability and application of non-volatile memory devices [J]. Journal of Integration Technology, 2022, 11(3): 42-55.

非易失性存储器件的性能、可靠性及应用

杜亚娟[1*] 金凯伦[1] 王子烨[1] 宁新杰[2]

[1](武汉理工大学计算机与人工智能学院 武汉 430070)
[2](武汉适库科技有限公司 武汉 430065)

摘 要 随着大数据和人工智能应用的发展，数据呈现爆发式增长，对数据存储的需求日益加剧。传统内存技术的容量已经接近其物理存储密度的极限，而非易失性存储器具有按字节寻址、能耗低、读写速度快等优良特性，有望替代传统的动态随机存储器或磁盘技术。然而，该介质本身也存在一些不足，如使用寿命有限、读写速度不对称、磨损不均衡和错误来源多样等缺点。该文通过阐述常见非易失性存储器的存储原理，调研并总结了一些现有改进技术。

关键词 非易失性存储器；磨损均衡；读写性能；内存压缩；可靠性
中图分类号 TP 391 **文献标志码** A doi: 10.12146/j.issn.2095-3135.20211017001

Performance, Reliability and Application of Non-Volatile Memory Devices

DU Yajuan[1*] JIN Kailun[1] WANG Ziye[1] NING Xinjie[2]

[1](*School of Computer and Artificial Intelligence*, *Wuhan University of Technology*, *Wuhan* 430070, *China*)
[2](*Wuhan Shiku Technology Co*., *Ltd*., *Wuhan* 430065, *China*)
[*]*Corresponding Author*: *dyj@whut.edu.cn*

Abstract With the development of big data and artificial intelligence applications, data are growing explosively, and the demand for data storage is increasing day by day. The capacity of traditional memory technology is approaching the limit of its physical storage density. Non-volatile memory is expected to replace traditional dynamic random access memory or disk technology due to its excellent characteristics such as byte addressability, low energy consumption, and fast read and write speed. However, the storage medium itself has some shortcomings, such as limited lifetime, asymmetric read and write speed, uneven wear and various sources of errors. The storage principles of common non-volatile memories are explained,

收稿日期：2021-10-17 **修回日期**：2021-12-23
基金项目：国家自然科学基金项目(61802287)；计算机体系结构国家重点实验室(中国科学院计算技术研究所)开放课题(CARCH201903)
作者简介：杜亚娟(通讯作者)，博士，副教授，硕士研究生导师，研究方向为计算机体系结构和存储系统，E-mail：dyj@whut.edu.cn；金凯伦，硕士研究生，研究方向为内存压缩算法；王子烨，硕士研究生，研究方向为 SSD 存储系统；宁新杰，硕士，研究方向为软硬件适配及性能优化。

and existing improved technologies are investigated and summarized.

Keywords　non-volatile memory; wear leveling; read and write performance; memory compression; reliability

Funding　This work is supported by National Natural Science Foundation of China (61802287), and State Key Laboratory of Computer Architecture (ICT, CAS) (CARCH201903)

1　引　言

随着大数据和人工智能时代的到来，数据呈现爆发式增长，对数据存储的需求日益加剧。现有的内存技术已逐渐趋于集成密度的极限，在现有技术基础上，无法持续扩展内存容量。此外，若要提升存储系统性能，须重点考虑现有外存技术与内存技术之间巨大的性能差异。非易失性存储器件(Non-Volatile Memory，NVM)的出现，使容量密度和性能差异的问题有望得到解决。非易失性存储是一种断电后数据不会丢失的新型存储技术。它具有可按字节寻址、能耗低、读写速度快等优良特性，被广泛研究的 NVM 有相变存储器(Phase Change Memory，PCM/PRAM)、自旋转移力矩存储器(Spin Torque Transfer Memory，STT-RAM)、阻变存储器(Resistive Memory，ReRAM)、铁电存储器(Ferroelectric Memory，FeRAM)等。这些存储器有望替代传统的动态随机存储器(Dynamic Random Access Memory，DRAM)内存技术。闪存作为固态硬盘的主要介质，通常也被归结为非易失性存储器的一种。

但是，非易失性存储器件还存在一些由介质本身引起的不足，例如：(1)使用寿命有限，当写操作达到一定次数后，存储的数据就不再可靠；(2)读写速度不对称，读速度往往大于写速度；(3)存储单元或存储块磨损不均衡，由于单元间的读写次数不同，引起磨损不均的问题；(4)当存储器损坏时，会卡在某个固定数值上，但由于存储器的材料限制，有些错误不能修复，因此，需要针对不同错误设计纠错或降低错误的算法。

本文首先阐述了几种常见的非易失性存储器的存储原理；其次，根据其特点对现有的一些改进技术进行讨论；再次，讨论它们在 3 种典型场景中的应用情况；最后，对非易失性存储器件的应用前景进行总结与展望。

2　非易失性存储器件的存储原理

本节将对几种 NVM 的存储原理进行介绍，主要包括 PCM、STT-RAM、ReRAM、FeRAM 和闪存存储器。

2.1　相变存储器

PCM 由加热器、硫化物和两个电极组成，底层电极连接加热器，加热器连接硫化物(一般为 $Ge_2Sb_2Te_5$[1])，硫化物再与顶层电极连接(如图 1 所示)。硫化物经过不同的加热过程会得到两个不同状态：当对需要写入的单元施加一个短却高的电压时，该硫化物会加热到非晶体的状态(600 ℃)，此时，硫化物的电阻率较高，该过程对应写 0(重置操作)；当对硫化物施加一个长却低的电压时，该硫化物会转化为晶体状态(300 ℃)，此时，硫化物的电阻率较低，该过程对应写 1(写 1 操作)。在两种状态之间，硫化物的电阻为 10^2～10^4[1]。由于硫化物在两个状态下的阻值差异较大，当选取其间不同的阻值时，就可以使用一个单元来存储多于一个比特的数值。如选取其间的 4 个电阻值，就可以用来表示 00、01、10、11 这 4 个数值。由于 PCM 存在电阻漂移现象，所以选取的电阻值越多，电阻值之间的差异就越小，就越易发生错误。

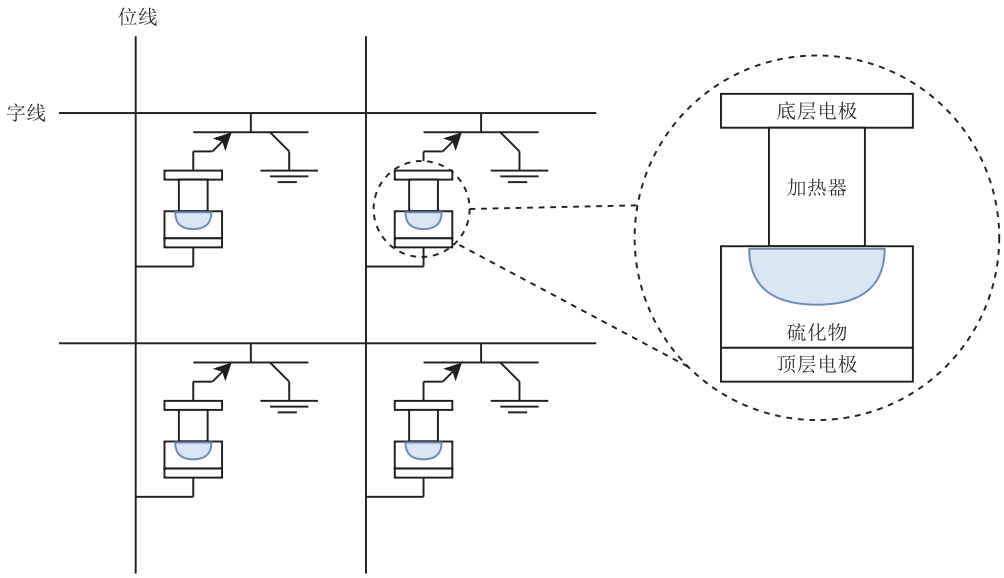

图 1 PCM 构造原理图[1]

Fig. 1 PCM structure schematic diagram[1]

与其他非易失性存储器相比，PCM 的单元最小、扩展性最好，但是其寿命却不是最好的。由于重置操作的高温会使相变材料的体积发生变化，连续发生膨胀或收缩会导致接触面不平，电极的接触面积变小(接触不好)。相变材料还可能与电极材料相互作用形成多余的化合物，从而导致相变材料被污染，缩短存储器的寿命，即写操作次数在 10^8～10^9 之间。而 PCM 的写入过程主要受写 1 操作的影响，当一个单元中存储多个逻辑值时，需要进行多次迭代，不同的迭代次数也会影响写入时间。PCM 的读延时在 20～60 ns 之间，写延时在 20～150 ns 之间[1]。虽然 PCM 的功耗相对较低，但是读写动态能量较高，尤其是写能量。

2.2 自旋转移力矩存储器

SST-RAM 由两个磁铁层和一个氧化物层组成，其中，两个磁铁层被氧化物层分开(如图 2 所示)。下方的磁铁层有固定的磁化方向，一般

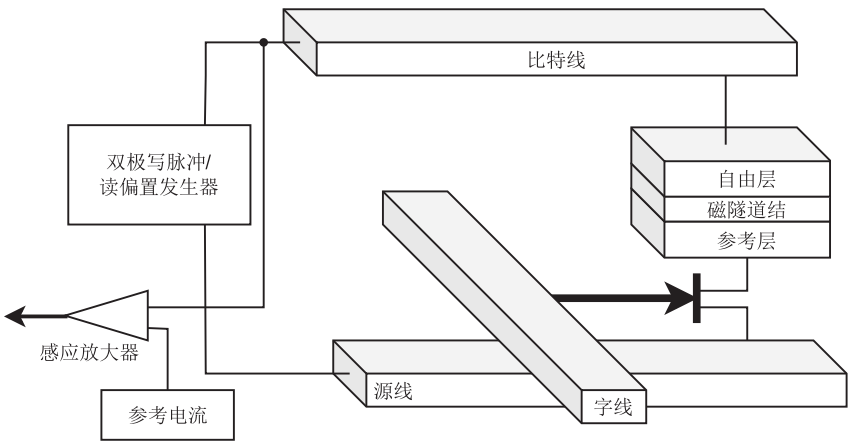

图 2 STT-RAM 构造原理图[1]

Fig. 2 STT-RAM structure schematic diagram[1]

将该层称为参考层；上方的磁铁层磁化方向可根据施加在比特线与源线之间的电压改变，称为自由层。当需要写入 0 时，在源线与比特线之间施加高正向电压，导致自由层和参考层的磁化方向相同，呈现低电阻率；当需要写入 1 时，在源线与比特线之间施加高负向电压，导致自由层和参考层的磁化方向相反，呈现高电阻率。

与其他技术相比，该技术更加成熟，且密度高、读写速度快、泄漏低、寿命长，写操作次数可达 10^{12}～10^{16}。后续的自旋轨道矩存储器(Spin Orbit Torque RAM，SOT-RAM)使用垂直翻转，并增加端分隔读取和写入操作的途径，使得读写速度更快(读延迟在 2～35 ns 之间，写延时在 3～50 ns 之间[1])。虽然 SOT-RAM 的静态能耗较低、读取时的动态能耗也低，但是写入时的能耗很高。

STT-RAM 被视为最有可能替代缓存的技术。但仍有两个不足之处，一是所使用材料具有热不稳定性，易造成数据丢失，这可能受连接、温度、从写入到最后一次读取的时间、内存单元数等因素的影响；二是高写入电流会降低交合处的接触程度，使得接触面发生变形，限制 STT-RAM 单元结构的完整性。在对该设备进行读取时，需要注入一定的电流，故会对设备磁性有所影响。

2.3　阻变存储器

ReRAM 属于忆阻器设备，忆阻器可以根据施加电压的大小、极性和持续时间改变它的阻值，在断电时其阻值不会发生改变。当忆阻器呈现低阻值时，认为此时存储的是 1；反之，存储的是 0。ReRAM 的单元由两个金属导体和夹在导体之间的绝缘体或电阻材料构成。其中，顶部和底部的两个金属导体一般使用 TiO_x 材料，也可使用如 ZrO_x 和 NiO 之类的材料。当施加电压会使单元中的绝缘体和电阻材料之间形成导电丝时，该单元为低电阻，即逻辑 1；当连接顶部和底部的导电丝中断时，该单元为高电阻，即逻辑 0。ReRAM 构造原理图如图 3 所示。

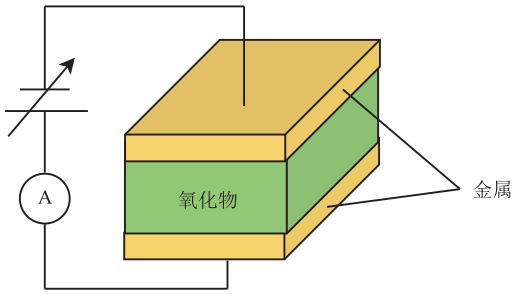

图 3　ReRAM 构造原理图[1]

Fig. 3　ReRAM structure schematic diagram[1]

ReRAM 的能耗比 PCM 低，尤其是写入时的能耗，集成的密度比 STT-RAM 高，且稳定性好，最大写入次数在 10^8～10^{11} 之间，其读取速度也很快，读取延迟近 10 ns，与 SRAM 相似，但其写入性能较差，写入延时近 50 ns[1]。

2.4　铁电存储器

组成 FeRAM 的材料是一个晶体管和铁电容器。其中，铁电容器是由两个金属电极和 $Pb(Zr_x,Ti_{1-x})O_3$(锆钛酸铅薄膜)组成，锆钛酸铅薄膜在两个电极之间，用于存储电荷。锆钛酸铅薄膜具有铁磁性，无须电荷即可保持极性，不同的极性可表示不同的逻辑值，故它具有非易失性。当写入 0 时，需要对板线施加强制脉冲；当写入 1 时，对位线施加强制脉冲。施加脉冲时，只须提供与电路供电电压相同的电压即可，可减小能耗。FeRAM 构造原理图如图 4 所示。

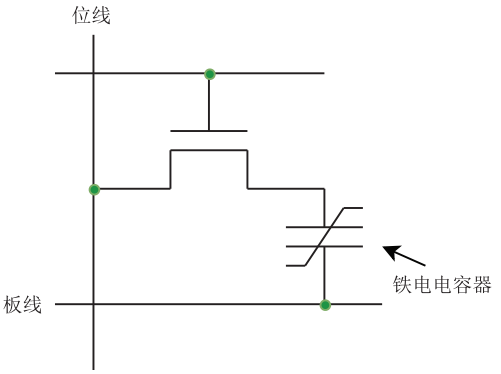

图 4　FeRAM 构造原理图[1]

Fig. 4　FeRAM structure schematic diagram[1]

FeRAM 比其他 NVM 提前进入量产，其读

取速度几乎和 DRAM 相同，在 20～80 ns 之间；写入速度较慢，在 50～75 ns 之间；能耗校低；不需要先擦除再进行写操作，且无须进行刷新操作，最大写入次数为 10^{14}～10^{15}[1]。由于该设备难以拓展，所以更适用于嵌入式设备。

2.5 闪存存储器

闪存存储器是所有 NVM 中最早投入使用的设备，也是最早研究、使用的设备。闪存存储器的存储单元由 PNP 型三极管组成，如图 5 所示。

理论上闪存存储器可读写的次数为 10^6，其中，与非门闪存的读写次数在 10^4～10^5 之间，读速度在 15～34 μs 之间，写速度在 200～500 μs 之间，动态能量和泄漏功率都很低[1]。

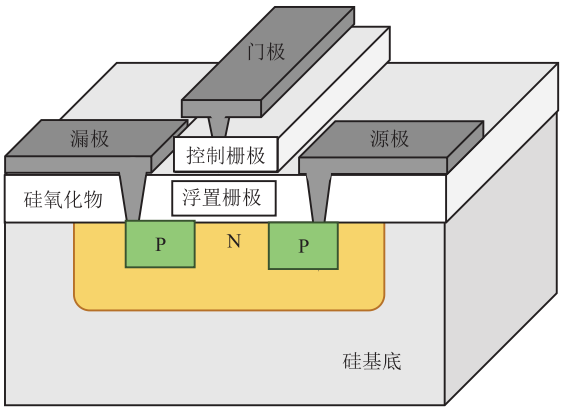

图 5 闪存存储器构造原理图[1]

Fig. 5 Flash structure schematic diagram[1]

3 非易失性存储器性能

由第 2 节可知，许多 NVM 的读写性能都具有不对称性——读取速度比写入速度快很多，且读写能耗也不同。所以针对 NVM 的读写不对称问题，有研究对 NVM 的读写性能进行了改进。NVM 的读写速度比传统外存快，所以也有研究将 NVM 集成到存储系统中，以提升存储器的整体读写性能。本节将对 NVM 的读写性能进行介绍。

3.1 基于存储介质特征的改进技术

增加性能最直接的方法是一次性读写多条缓存行，但是一味地增加带宽不仅会产生大量能量消耗，而且会增加硬件成本，压缩/压实能很好地解决这一问题。Palangappa 等[2]发现多级单元(一个单元里存放两个比特)/三级单元(一个单元里存放 3 个比特)NVM 编程需要进行迭代，造成了高能耗和高延迟，故提出压缩扩展编码技术。该编码技术将频繁模式压缩(Frequently Pattern Compression，FPC)/基数-偏移量-立即数压缩与 $(k,m)_q$ 扩展码技术相结合，选择性地将压缩后的数据进行拓展，确保拓展后的数据对应 MLC/TLC 单元的低编码能量，以减少能量消耗和延时。$(k,m)_q$ 扩展码将一组占据 m 个存储单元的信息(原始信息)映射为一组占据 k 个存储单元的信息(映射后信息)，k 大于 m。其中，原始信息中每个单元可以表示 q 个状态，映射后的信息只能表示 q 个状态中 p 个能量较低的状态。

Chen 等[3]将 NVM 作为块设备使用。他们发现使用压缩技术可以减少成本，但会增加延迟，而 NVM 具有细粒度读写这一优势，可根据这一优势优化解压缩、截断纠错码(Error Correction Codes，ECC)、基于数据访问位置的压实、页内增量编码 4 种技术。优化解压缩就是对压缩与 BCD 的解码完整性检查并行执行，以隐藏解压缩延迟。在 NVM 中，单个页面受到单个长 ECC 码的保护，但在访问数据时，需要先根据 ECC 码对数据进行解码或者编码，导致 NVM 无法进行细粒度的数据访问，这消除了压缩带来的好处，为此截断 ECC 被提出。截断 ECC 就是截断 ECC 中的无效部分(假设系统使用 n 位 ECC 码保护 k 位的数据，但是使用 n 位 ECC 码的前 m 位也能保护 k 位数据，那么后面的 $n-m$ 位就是无效部分)，使 NVM 能进行细粒度读写，从而减少延迟。Chen 等[3]发现压实过程中的数据打包和垃圾回收会增加性能开销，所以提出利用数据访问的局部性，只将冷数据打包得较为紧凑，而热数据打包得相对不那么紧凑。Chen 等[3]还发现，数

据块的重写有很强的相似性，而细粒度的写可以减少写的范围，因此，提出了页内增量编码。

Guo 等[4]利用数据分布特性获取的动态模式来进一步加大压缩率，并借助多级单元/三级单元 NVM 的不同数值写入特性改进写入机制。利用该方法对运行时提取到的模式进行分析，得到最常用的模式，将这些模式替换 FPC 的静态模式，可提高压缩算法的压缩率。此外，该方法还拓展了数据模式的大小，即加大单个压缩的粒度，以减少全 0 行的写入比特数，并根据拓展模式选择合适的压缩算法。对于三级单元 NVM，该方法在一个单元中存储两比特有效数据，并在每两比特有效数据的最高位前添加 0 或 1，从而扩展为三比特数据。加 0 还是加 1 取决于扩展后的三比特数据所需的编程能量大小，选取能量较小的三比特数据作为扩展数据，由此减少写入能量。

但也有一些技术会产生额外的延迟。低密度奇偶校验(Low-Density Parity-Check，LDPC)是一种被广泛应用于闪存以提高纠错能力的技术，然而，LDPC 会导致读取性能下降。有研究[5-8]针对 LDPC 进行改进，通过减少解码时间来提升性能。Du 等[9]发现 LDPC 的读取级别与延迟成正比，提出了一种多粒度 LDPC 读取方法，来适应每一层读取级别的增加速度。Du 等[10]考虑将用于提高闪存寿命的刷新概念用来优化闪存读取性能。该方法利用数据读取特性，提出了一种轻量级数据刷新方法轻量级数据刷新(Lightweight Data Refresh，LDR)——可以积极纠正读取延迟较长的读取热点页面中的错误，并将无错误数据重新编程到新页面中。Du 等[11]为了避免不必要的重试读取操作提出了延迟感应的低密度奇偶校验(Latency-aware Low-Density Parity-Check，LaLDPC)。由于在 SSD 中会在相当长的一段时间内都使用相同的读等级，在此期间，同一页中的所有读取具有相同的读取级别，LaLDPC 可利用读取级别估计起始读取电压电平，并将其存储在控制器的闪存转换层中。Du 等[11]还提出一种新的缓存驱逐算法，以尽可能长时间地将具有高读取级别的条目留在缓存中。还有一些研究[12-14]针对垃圾回收算法进行改进，通过减少垃圾回收时某些操作的时间，来提升性能。

以上研究表明，当使用不同的技术时，会伴随多种性能降低的可能，如压缩带来的额外数据读取和解压缩延迟。当 NVM 作为内存使用时有容量和持久性的优势，内存缺页的情况会减少，可以缩短下一级向内存传输数据的时间；NVM 作为外存时可以进行细粒度访问和达到较快的读写速度，来抵消压缩带来的负面影响。

3.2　存储器架构集成技术

由于 NVM 的性能介于 DRAM 内存与外部存储器之间，所以将 NVM 集成到存储体系结构中可提升存储系统整体的性能[15-18]。Korgaonkar 等[15]发现增加最后一级缓存的容量，可减少 CPU 和内存之间的性能差距，所以将 STT-RAM 作为最后一级缓存，并设计了写拥塞感知绕过和虚拟混合混存来消除大部分的正常写和全部的冗余写。Kommareddy 等[16]考虑到现在的应用对内存的需求很大，而 NVM 的速度不如传统内存，故设计了一个分片式 NVM 内存——在每个计算节点中设置小型的基于高带宽存储器或者 DRAM 的局部内存，而这些节点共享一个大的 NVM 内存。在分片式 NVM 的基础上，系统管理方法也进行了优化，优化后会将尽可能多的热数据停留在局部内存中，减少数据迁移时的冲突。Xu 等[19]研究了速度变化对 SSD 并行性能的影响，并建立一种新的具有闪存层信息意识的排队时间估计模型，该模型评估 SSD 各个芯片的排队时间，从而将请求定向到排队时间最少的芯片。

Patil 等[20]比较了 4 种不同的 DRAM-NVM 混合内存系统，其中，NVM 使用的是基于 PCM 的设备。第一种是将 NVM 作为主存，并未用到 NVM 的非易失性；第二种是将 NVM 作为

持久性存储设备，DRAM 作为主存；第三种将 DRAM 和 NVM 作为混存，一部分 NVM 作为主存，而 NVM 剩下的部分作为持久性存储设备；第四种将 NVM 与 DRAM 都作为内存，并统一寻址。最终发现，当 DRAM 和 NVM 一起作为混存使用时，既可以保证 DRAM-NVM 混合内存系统具有与传统系统一样的性能，又可以增加内存容量。当 NVM 作为主存时(例如第一种和第三种情况)，虽然会降低读写速度，但是能够大量减少能量消耗。

Park 等[21]提出了一种基于多分区新内存控制器和子系统：将 DRAM 与 PCM 进行集成，其中 DRAM 作为缓存以隐藏底层 PCM 模块所施加的长延时，且支持持久性操作，分区的设计提供了非阻塞读服务(允许在写入内存的同时进行多个读取操作，以减少读写延迟)。Li 等[22]将多级单元 PCM 作为块设备集成到系统中，多级单元 PCM 有两种写入操作，一种是快速但易失的写入，另一种是慢速但非易失的写入。该文基于这两种写入模式提出了一种编译器定向双写方案。该方案首先对每个内存写入指令的寿命(从该指令写入内存到最后一次读取所经过的时间)进行分析，然后根据分析结果选择写入模式，并根据写入模式将指令插入要编译的代码，为每个内存写入指令选择最佳写入模式，以权衡写入延时和停留时间。

Zhang 等[23]提出了一种原始误码率感知的多传感方案，该方案可用于减少传感和传输延迟，从而提高读取性能。Li 等[24]发现选择性地降低电压可以减少读取干扰，所以提出了一种读取热度感知降低电压的方案，该方案在不违反可靠性要求的前提下，可提高存储器性能。

上述研究表明，无论是将 NVM 集成作为混存还是主存，都会面临性能上的挑战，在设计系统时，如何利用混合内存或者混合缓存避免增加延时是关键。此外，在设计系统时，可以考虑利用 NVM 的非易失性、能耗低、易于集成等优点来减少系统的性能开销。

4 非易失性存储器寿命及可靠性

如第 1 节所述，每次写入时，NVM 都会产生磨损，所以 NVM 的写入次数有限，这是一个急需解决的问题。本节将从减少比特翻转、减少写入次数、纠错码、磨损均衡和其他技术这 5 个方面进行讨论。

4.1 减少比特翻转

根据不同 NVM 的写入原理可以发现，每次写入都会对构成 NVM 的写入单元造成破坏，所以在单次写入时，若能减少对每个单元的比特翻转(置 0 或者置 1 操作)，就可以增加 NVM 的寿命。Cho 等[25]提出的 Flip-N-Write 主要利用“读-修改-写”替换写操作和选择性翻转。“读-修改-写”操作可以在旧数据和新数据的位逻辑值相同时，跳过写入。在进行选择性翻转时，统计在“读-修改-写”操作后写入新数据时必须翻转的位的数量，当翻转的位数超过存储器字宽的一半时，就翻转要写入的新数据，这样位翻转的数量就永远小于字宽的一半。改变字宽可以在不同程度上减少位翻转，字宽越小，减少效果越好。Chen 等[26]提出了一种压缩技术——基于算术编码的技术，用合适的浮点数替换需要存储的数据，有效地减少了比特翻转。即先统计写入中不同字符出现的次数，计算出现的概率，随后在[0,1)区间内寻找概率相应的子区间，只需从子区间选择一个浮点数表示该输入即可。由此，算法可以从中挑选与旧数据相比翻转位数最少的浮点数来替换输入的新数据，从而减少比特翻转。

Dgien 等[27]提出的架构使用 FPC 压缩算法、一个比较器来减少比特写入。首先使用 FPC 对写入的数据进行压缩的同时，内存控制器会读取当前位于内存中目标地址处的旧数据。然后当压

缩完成时，比较器逐位对新旧数据进行对比，以确定数据会发生翻转的比特，写入时只更新发生翻转的比特的位置，以此减少位的写入。Kargar 等[28]提出将 DRAM 与 NVM 集成作为混合内存的汉明树，这是一种使用现有索引进行扩充的辅助数据结构，可用于任何一种基于树的数据结构。该结构将两种内存映射到单个物理地址空间上，寻找与要写入的新数据最相似的旧数据的位置，将写操作定向到内存位置，从而最大限度地减少位翻转次数。

Bittman 等[29]分析了减少位翻转对 NVM 性能的影响，并根据这些影响对异或链表、异或哈希表、异或红黑树这几种减少位翻转的技术进行测试，并作讨论。Bittman 等通过研究发现，减少位翻转可以对数据结构设计、程序操作、混存层产生影响。异或链表是一种双向链表设计，每个节点不存储前一个节点和下一个节点的位置，而是存储前一节点位置和下一节点位置的异或(头尾节点存储相邻节点完整的指针值)，可以利用异或结果减少位翻转。异或哈希表是将异或链表运用到哈希的同列表中，此时运用的异或链表不存储上一个节点和下一个节点的异或值，而是存储当前节点和下一个节点的异或值。此外，异或哈希表还允许下一个指针的最低有效位为 1 或者将数据指针设置为空来标记空链表。异或红黑树与异或链表使用相同技术来减少位翻转。

已有的减少位翻转的技术都是先利用“读-修改-写”技术，再通过改变写入的数据或位置以减少位翻转次数。如使用压缩、扩展数据、翻转数据、对特殊结构进行特定操作等技术改变写入的数据。或许还可以利用其他技术对数据进行改变，从而减少位翻转。鉴于 MLC NVM 的写入特性，也有研究甚至忽略位翻转数量，而是选择几个能量低的状态进行写入，以减少写入能量。

4.2　减少写入次数

除减少每次写入时需要修改的比特数之外，减少整体的写入次数也是一种增加 NVM 寿命的方法。在 NVM 进行集成时，许多研究都使用了缓存技术和写回技术，当中央处理器发出存储指令时，指令中要存储的数据不会直接写入 NVM，而是先将数据写回缓存，等缓存溢出时，再写回 NVM。由于数据的局部性，许多写入都是对同一个缓存行进行的，所以在该缓存行被驱逐出缓存前，所有对该缓存行的写入都会在缓存中进行合并，这样可大量减少 NVM 的写入次数。Qureshi 等[30]提出了将 DRAM 与 NVM 集成作为混合内存的模型：将 DRAM 作为 NVM 的缓冲，负责存储热页，当触发缺页时，处理程序只把获取的页面发送到 DRAM。只有当页面从 DRAM 中被驱逐时，才会被写入 NVM 中，并且该模型还在 DRAM 与 NVM 之间增加了写缓存，当数据从 DRAM 写入 NVM 时，该模型使用缓存行粒度，仅将修改过的缓存行写回 NVM，从而进一步减少了写次数。此外，还设计了页面级的绕过技术，当页面重用性很差时，不会被写入 NVM。

Ni 等[31]发现，为了保证正常关机后系统的一致性而使用的日志记录和影子分页技术，都会记录同一页面中不进行操作的缓存行，会对 NVM 造成额外的写，所以提出了优化的影子分页技术——利用虚拟内存的间接性来避免记录实际数据。该技术在每个有效页面都使用紧凑的缓存行级别的映射，使每个虚拟页面与两个物理页面相关联，当进行写入时，其中一个物理页面存储的是相关联的虚拟页面修改前的一致状态，而另一个物理页面存储的是相关联的虚拟页面在缓存行修改后的状态。优化后的影子分页技术，会大大减少额外写操作。

Zhang 等[32]发现多级单元相变存储器会出现电阻漂移现象，不同的写入次数会产生不同的延迟和不同的停留时间。迭代次数越少，延时越短，出现电阻漂移的时间越短，停留时间也越

短。对于频繁写入的区域而言，更新数据的时间短、次数多，所以仅需数据短时间的停留和快速地写入。基于此，该文章分区域对迭代次数进行了研究，对于频繁写入的区域进行迭代次数较少的写入，对于不频繁写入的区域进行迭代次数较多的写入。

减少写入次数的方法多种多样，合并写入就是常见的一种。除了减少正常写入，有时系统和运行的程序中也会存在不需要的写入，如果能探测到这种写入并删除，就可以减少写入次数。

4.3 纠错码技术

对于已经出错的单元，使用 ECC 技术能够在一定错误率范围内进行修正，从而使这些出错的单元能够重用，延长 NVM 的寿命，最常用的为 BCH 码。若要检验的错误数量为 n，BCH 码需要 1 个不会被分解的多项式(本原多项式 $p(x)$)和 $(n-1)$ 个其余多项式($p_3(x), p_5(x)\cdots p_{2n-3}(x)$)，其余多项式在赋值 x^n 后要能整除本原多项式，n 对应其余多项式的底标，即 $p_n(x^n)$ 对本原多项式取余为 0，将本原多项式和其余多项式相乘就能得到编码多项式。在发出信息时，发送的信息需要乘以编码多项式，然后将相乘后的结果发出，需要检验时，将接收到的信息除以本原多项式，若余数为 0，就表明没有出错。根据余数的状态，进行一系列较为复杂的运算，就可以检测出具体出错的地方。奇偶检验也是较常用的一种技术，它利用冗余位(会根据数据位的数量而改变)进行纠错。奇校验就是检查发送信息中 1 的个数，当 1 的个数为奇数时，就将冗余位设置为 0，否则设置为 1。汉明码作为奇偶校验的升级版，在编码后的数据中，2 的幂次位上的数据为奇偶校验位，其根据数据的二进制索引位置，将数据与奇偶校验位进行对应，奇偶校验针对分到数据中 1 的个数置 1 或置 0。在检验错误时，只需要再次对数据进行奇偶校验，就可得到错误的位置。但是，使用 ECC 技术会带来一些开销，所以有研究利用 NVM 的特性对 ECC 进行改进。

Kwon 等[33]发现多级单元 NVM 由于电阻漂移会引入许多软错误，所以需要 ECC 来进行纠正。该文提出了一个可靠的多级单元 PCM 架构，其利用电阻漂移产生的数据相关性来减少 ECC 开销，并与 BCH 编码技术相结合。基于多级 PCM 电阻漂移的特点——电阻漂移引起的状态变化只会有一比特，且能确定是由什么状态漂移成的什么状态(通常电阻高的漂移称作比它电阻低一级的状态)，该架构生成 BCH 码时只需要考虑其中一位。该架构使用简单的 2 比特到 1 比特状态映射生成虚拟数据，生成的数据大小仅为原始数据的一半，将生成的数据用于校验，减少了由 BCH 码产生的附加位。在进行数据读取时，BCH 码会检验数据，当发现错误时，就可根据漂移的属性来进行还原。

Lu 等[34]发现使用 ECC 会产生大量的存储空间，提出了渐进式 ECC 技术。该技术只为每个数据字配备奇偶校验位以检测故障，当检查到数据字中第一个错误位时，才会对数据字配备 ECC，并使用纠错技术修复错误位，然后在 ECC DRAM 中分配该校验条目。Kim 等[35]发现当重复读取 PCM 单元时，会发生读取干扰，常规的解决方法是定期清理单元，但这种方式需要读取计数器来计算每个字的读取次，较为费时。针对常规解决方法的缺陷，该文提出了一种按需清理技术，不需读取计数器，可利用 ECC 来观察单词中发生错误的次数，当且仅当错误的次数大于阈值时，才执行清理来修复错误。

与其他技术相比，ECC 技术发展得较为完善，可在 SSD 中运用的 ECC 相关技术也可直接用于 NVM。对 NVM 而言，ECC 的问题就是占用内存较多，或许可以考虑结合压缩技术解决该问题。

4.4 磨损均衡技术

在一个存储部件中，不同单元的写入次数是

不一样的，这取决于该单元中存储数据的热度。存储热数据的单元由于读写次数明显高于存储冷数据的单元，所以受到的磨损程度更大，损坏也越快。当损坏块超过一定数量时，该存储器随之损坏。磨损均衡技术就是尽可能地平均每一个单元的损坏程度，从而避免有的单元已经损坏，但是仍有单元进行读写的次数很少。磨损均衡算法的大体思想就是避免一直对同一个地址进行数据写入，该算法一般与压缩技术相结合。Dgien 等[27]提出利用机会主义磨损均衡器来减少 NVM 单元的峰值位写入，该方法通过压缩技术获得额外空间，并有条件地将压缩数据写入 NVM 整列字的另一端来减少磨损。

Liu 等[36]提出了一种基于空间遗忘压缩和磨损均衡的内存框架，通过压缩后的可用内存空间来实现块内磨损均衡策略。该框架在数据的初始存储空间内，轮换压缩数据块的写入位置，从而均衡每个 NVM 单元的写入。其首先将内存块均匀地划分为 4 个部分——“00”、“01”、“11”、“10”，然后利用地址旋转算法从不同部分的起始字节开始存储，从而达到均衡磨损的目的。具体存储的开始位置由是否压缩、前一个地址标签和压缩数据的大小决定。该算法的原理是在上述 4 个开始地址之间迭代旋转地址标签，为了减少磨损，迭代的顺序为 00-01-11-10。

4.5　其他技术

除上述方法外，还有一些可增加 NVM 设备寿命的方法。Soltani 等[37]提出了一种用于容忍 PCM 存储加密数据时卡住故障的方法。该方法利用由高级加密标准(AES)编码的加密数据的随机特性以及旋转移位操作，可使大量具有固定故障的存储位置来正确存储数据。一般地，若 PCM 已经磨损，那么就会永久停留在 0 或 1 上，而 AES 数据之间的随机性很高，数据经 AES 加密后，就能将数据进行循环移位或翻转，大概率地匹配已经磨损单元的值。若生成的加密数据无法与磨损单元进行匹配，也可以重新生成加密数据。

Liu 等[36]发现许多应用程序的内存块通常包含大量的 0 字节和频繁值，所以提出先将都是 0 的重复数据删除，再进行频繁值压缩，以减少 NVM 单个单元上的写入次数。0 重复数据删除就是使用比特图去编码和定位全是 0 的内存块，当对应块全是 0 时，就将比特图中对应的位置置为 1。频繁值压缩就是找到内存中应用较频繁的值，利用值码对应表对其进行编码(编码长度小于值长度)，当进行解压缩时，就从值码对应表中寻找编码对应的值，从而进行还原。

5　产业应用

NVM 已经被运用在多种应用场景中。本节将对 NVM 的 3 种应用场景进行介绍，分别是日志结构合并树(Log Structured Merge Tree，LSM-tree)、神经网络和语意事务。

5.1　在日志结构合并树上的应用

Yao 等[38]使用 DRAM-NVM-SSD 的存储结构，根据 NVM 的特性，对基于 LSM-tree 的键值存储进行了改进。LSM-tree 可存储多级的键值项，每一级的数量呈指数增加。但是，基于 LSM-tree 的键值存储存在两个明显的缺陷，一是在 L_0 和 L_1(最上面两层)之间存在写入停顿的现象，二是写放大会随着 LSM-tree 深度的增加而增加。写入停顿会导致应用层序吞吐量周期性地下降，直至接近于 0，导致长尾延时，降低用户体验感。产生写入停顿的主要原因是 L_0 没有排序，所以 L_0 与 L_1 之间的压缩会涉及两层的所有数据。针对该问题，Yao 等[38]提出了 4 种新的技术：(1)将 L_0 从 SSD 移至 NVM 中，提出用矩阵容器重新定位和管理 NVM 中的 L_0，由于 NVM 具有按字节寻址和快速随机访问的能力，而改进后的行表使用 NVM 页作为基本单元，所以该结

构允许 L_0 和 L_1 之间更细粒度的数据迁移；(2)设计了新的列压缩，通过在细粒度的键范围内压缩 L_0 至 L_1 的数据，减少压缩的数据量；(3)增加每个级别的宽度，以减小 LSM-tree 的深度，从而减小了写入放大；(4)针对矩阵容器引入跨行提示搜索，即在 L_0 对行表进行排序，不同的行表使用不同的键范围进行覆盖，在构建行表时，为元数据排序数组中的每个元素添加向前指针，以保证足够的读取性能。改进本身结构后，利用 NVM 的非易失性，将写前日志写入 NVM，数据写入 DRAM。当读取数据时，由于 NVM 比传统的外存设备的读写速度更快，所以读取数据的时间也更短。Hu 等[39]针对压缩给 LSM-tree 带来的尾延迟进行改进分析，提出有限压缩，即只允许部分层级参与压缩来改善性能。

5.2 NVM 在神经网络上的应用

Yu 等[40]总结了利用 NVM 设备进行神经启发计算的最新技术、挑战和前景。神经启发架构能够有效解决片上存储和片外存储之间显著的性能差异。神经网络的训练分为离线训练(使用软件进行训练，利用一次性编程，将训练好的权重加载到神经形态硬件的突触阵列中，然后仅在硬件上进行推理或分类)和在线训练(运行时，在神经形态硬件上完成训练，权重是动态的)。NVM 的集成密度高，利用它替代 SRAM，可将更多权重存储在片上，从而减少对片外存储器的访问。利用 NVM 的电阻交叉杆替换 SRAM 阵列，还可进行并行编程和加权求和，使存储进一步加速，从而有可能实现在线训练。PCM 在 0 和 1 两种状态下的电阻差异较大，所以可以充当模拟突触，该模型存在的问题是重置操作过程快，难以控制。由于 ReRAM 的写 1 和写 0 操作与 PCM 相反，所以写操作需要渐进。FeRAM 组成的突触件是一种三端结构，可将权重调谐和权重读取路径解耦，与 ReRAM 的模拟突触器件相比，它有更大的开/关比、更短的编程脉冲宽度，以及更小的权重更新曲线变化。

5.3 结合语义事务的应用

Ramalhete 等[41]发现当出现非破坏性故障时，将事物语义与持久技术相结合，数据能始终保持在持久性内存中。Ramalhete 等[41]对栅栏、刷新的数量和内存使用进行了权衡，提出两种新算法(Trinity 和 Quadra)用于持久性内存上的持久事物。Quadra 将持久性栅栏数量的下限变低，并为每个修改的缓存线执行一条刷新指令。Trinity 为每个事物执行两条 Fences 操作，该算法易与基于细粒度锁定的并发控制技术相结合，将其与事务锁 II 算法适配集成在一起，可实现急切锁定和直写更新策略。此外，将 Trinity 和事务锁 II 算法组合到持久性事物内存中，从而为具有不相交访问模式的数据结构和工作负载提供了良好的可扩展性，并能对持久化线性事务实现键值存储。

6 总结与展望

通过相关调查研究发现，NVM 技术的应用前景十分广阔。然而，由于 NVM 的介质特性、可靠性、持久性等特性，直接使用 NVM 或将其与其他存储设备进行集成时，仍存在一定挑战。此外，现有应用大多基于传统存储器设计，更适用于传统存储系统。如何基于 NVM 的特性，设计适用于该硬件的应用或软件系统，是当前亟须考虑的问题。本文阐述了 NVM 的技术背景、现有优化工作和应用前景，为促进 NVM 的广泛应用和量产提供了强有力的支撑。

NVM 相关技术的未来发展趋势主要总结为 3 个方面：(1)新型或改进后的 NVM 技术将大量涌现。在具体应用中，由于现有的 NVM 技术仍存在一些不足，其发展之路任重道远。(2)结合 NVM 设备特点，应用应针对 NVM 的特有访问模式进行革新。在读写延迟和功耗等方面，由于新型 NVM 技术与传统存储技术存在较大差异，

改进应用访问模式或将极大挖掘 NVM 技术潜力。(3)存储架构的革新。当 NVM 技术发展到一定阶段后，其性能和存储容量优势极大凸显，存储架构也将进一步突破现有的冯·诺伊曼系统架构，如内外存一体化技术、存算一体技术和内存计算技术等新型存储架构将取得进一步的发展。

参 考 文 献

[1] Boukhobza J, Rubini S, Chen RH, et al. Emerging NVM: a survey on architectural integration and research challenges [J]. ACM Transactions on Design Automation of Electronic Systems, 2017, 23: 1-32.

[2] Palangappa PM, Mohanram K. CompEx＋＋: Compression-expansion coding for energy, latency, and lifetime improvements in MLC/TLC NVMs [J]. ACM Transactions on Architecture and Code Optimization, 2017, 14: 1-30.

[3] Chen XB, Li Y, Hao JP, et al. Simultaneously reducing cost and improving performance of NVM-based block devices via transparent data compression [C] // Proceedings of the International Symposium on Memory Systems, 2019: 331-341.

[4] Guo YC, Hua Y, Zuo PF. A latency-optimized and energy-efficient write scheme in NVM-based main memory [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 39: 62-74.

[5] Wu F, ZhangM, Du YJ, et al. Using error modes aware LDPC to improve decoding performance of 3-D TLC NAND flash [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2019, 39: 909-921.

[6] Zhang M, Wu F, Du YJ, et al. Pair-bit errors aware LDPC decoding in MLC NAND flash memory [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 38: 2312-2320.

[7] Wu F, Zhang M, Du YJ, et al. A program interference error aware LDPC scheme for improving NAND flash decoding performance [J]. ACM Transactions on Embedded Computing Systems, 2017, 16: 1-20.

[8] Zhang M, Wu F, Du YJ, et al. CooECC: a cooperative error correction scheme to reduce LDPC decoding latency in NAND flash [C] // Proceedings of the 2017 IEEE International Conference on Computer Design, 2017: 657-664.

[9] Du YJ, Zhou Y, Zhang M, et al. Adapting layer RBERs variations of 3D flash memories via Multi-Granularity progressive LDPC reading [C] // Proceedings of the 54th Design Automation Conference, 2019.

[10] Du YJ, Li Q, Shi L, et al. Reducing LDPC soft sensing latency by lightweight data refresh for flash read performance improvement [C] // Proceedings of the 2017 54th ACM/EDAC/IEEE Design Automation Conference, 2017: 1-6.

[11] Du YJ, Zou DQ, Li Q, et al. LaLDPC: Latency aware LDPC for read performance improvement of Solid State Drives [C] // Proceedings of the 33rd International Conference on Massive Storage Systems and Technology, 2017: 1-11.

[12] Du YJ, Liu W, Zhang Y, et al. United SSD block cleaning via constrained victim block selection [C] // Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, 2019: 250-257.

[13] Wu F, Zhou JN, Wang SZ, et al. FastGC: accelerate garbage collection via an efficient copyback-based data migration in SSDs [C] // Proceedings of the 2018 55th ACM/EDAC/IEEE Design Automation Conference, 2018: 1-6.

[14] Du YJ, Jia YP, Zhang M, et al. Enhancing SSD performance with LDPC-aware garbage collection [C] // Proceedings of the 2017 IEEE 6th Non-Volatile Memory Systems and Applications Symposium, 2017: 17244392.

[15] Korgaonkar K, Bhati I, Liu H, et al. Density tradeoffs of Non-Volatile Memory as a

replacement for SRAM based Last Level Cache [C] // Proceedings of the 2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture, 2018: 315-327.

[16] Kommareddy VR, Hammond SD, Hughes C, et al. Page migration support for disaggregated non-volatile memories [C] // Proceedings of the International Symposium on Memory Systems, 2019: 417-427.

[17] Song WN, Zhou Y, Zhao MY, et al. EMC: energy-aware morphable cache design for non-volatile processors [J]. IEEE Transactions on Computers, 2018, 65: 498-509.

[18] Liu K, Zhao MY, Ju L, et al. Applying multiple level cell to non-volatile FPGAs [J]. ACM Transactions on Embedded Computing Systems, 2020, 19: 1-22.

[19] Xu JM, Du YJ, Ding C. Layer-aware request scheduling for 3D flash based SSDs [J]. IEEE Access, 2021, 9: 72025-72032.

[20] Patil O, Lonkov L, Lee J, et al. Performance characterization of a DRAM-NVM hybrid memory architecture for HPC applications using intel optane DC persistent memory modules [C] // Proceedings of the International Symposium on Memory Systems, 2019: 288-303.

[21] Park G, Kwon M, Mahapatra P, et al. BIBIM: a prototype multi-partition aware heterogeneous new memory [C] // Proceedings of the 10th USENIX Workshop on Hot Topics in Storage and File Systems, 2018.

[22] Li QG, Jiang L, Zhang YT, et al. Compiler directed write-mode selection for high performance low power volatile PCM [C] // Proceedings of the 14th ACM SIGPLAN/SIGBED Conference on Languages, Compilers and Tools for Embedded Systems, 2013: 101-110.

[23] Zhang M, Wu F, Chen XB, at el. RBER aware multi-sensing for improving read performance of 3D MLC NAND flash memory [J]. IEEE Access, 2018, 6: 61934-61947.

[24] Li Q, Shi L, Di YJ, et al. Improving read performance via selective vpass reductionon high density 3D NAND flash memory [C] // Proceedings of the 2017 IEEE 6th Non-Volatile Memory Systems and Applications Symposium, 2017: 17244409.

[25] Cho SY, Lee HJ. Flip-N-Write: a simple deterministic technique to improve PRAM write performance, energy and endurance [C] // Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, 2009: 347-357.

[26] Chen YS, Wu CF, Chang YH, et al. A write-friendly arithmetic coding scheme for achieving energy-efficient non-volatile memory systems [C] // Proceedings of the 26th Asia and South Pacific Design Automation Conference, 2021: 633-638.

[27] Dgien DB, Palangappa PM, Hunter NA, et al. Compression architecture for bit-write reduction in non-volatile memory technologies [C] // Proceedings of the 2014 IEEE/ACM International Symposium on Nanoscale Architectures, 2014: 51-56.

[28] Kargar S, Nawab F. Hamming tree: the case for memory-aware bit flipping reduction for NVM indexing [C] // Proceedings of the 11th Annual Conference onInnovative Data Systems Research, 2021: 10-13.

[29] Bittman D, Long DDE, Alvaro P, et al. Optimizing systems for byte-addressable NVM by reducing bit flipping [C] // Proceedings of the 17th USENIX Conference on File and Storage Technologies, 2019: 17-30.

[30] Qureshi MK, Srinivasan V, Rivers JA. Scalable high performance main memory system using phase-change memory technology [C] // Proceedings of the 36th Annual International Symposium on Computer Architecture, 2009: 24-33.

[31] Ni YJ, Zhao JS, Bittman D, et al. Reducing NVM writes with optimized shadow paging [C] // Proceedings of the 10th USENIX Workshop on Hot Topics in Storage and File Systems, 2018.

[32] Zhang MZ, Zhang LK, Jiang L, et al. Balancing performance and lifetime of MLC PCM by using a Region Retention Monitor [C] // Proceedings of the 2017 IEEE International Symposium on High Performance Computer Architecture, 2017: 385-396.

[33] Kwon T, Imran M, Yang JS. Cost-effective reliable MLC PCM architecture using virtual data-based error correction [J]. IEEE Access, 2020, 8: 44006–44018.

[34] Lu SK, Li HP, Miyase K. Progressive ECC techniques for phase change memory [C] // Proceedings of the 2018 IEEE 27th Asian Test Symposium, 2018: 161-166.

[35] Kim M, Lee HJ, Kim H. An on-demand scrubbing solution for read disturbance error in phase-change memory [C] // Proceedings of the 2020 International Conferenceon Electronics, Information, and Communication, 2020: 1-2.

[36] Liu HK, Ye YY, Liao XF, et al. Space-oblivious compression and wear leveling for non-volatile main memories [C] // Proceedings of the 36th International Conference on Massive Storage Systems and Technology, 2020.

[37] Soltani M, Kamal M, Afzali-Kusha A, et al. RandShift: an energy-efficient fault-tolerant method in secure nonvolatile main memory [J]. IEEE Transactions on Very Large-Scale Integration Systems, 2019, 28: 287-291.

[38] Yao T, Zhang YW, Wan JG, et al. MatrixKV: reducing write stalls and write amplification in LSM-tree based KV stores with matrix container in NVM [C] // Proceedings of the 2020 USENIX Annual Technical Conference, 2020: 17-31.

[39] Hu YC, Du YJ. Reducing tail latency of LSM-tree based key-value store via limited compaction [C] // Proceedings of the 36th Annual ACM Symposium on Applied Computing, 2021: 178-181.

[40] Yu SM. Neuro-inspired computing with emerging non-volatile memory [J]. Proceedings of the IEEE, 2018, 106: 260-285.

[41] Ramalhete P, Correia A, Felber P. Efficient algorithms for persistent transactional memory [C] // Proceedings of the 26th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, 2021: 1-15.