通过消除读扰动累积提升自旋转移力矩磁阻随机存取存储器缓存的可靠性

1st ElhamCheshmikhani 2nd Hamed Farbeh计算机工程系 计算机工程与信息技术系 谢里夫理工大学 阿米尔卡比尔理工大学伊朗德黑兰 伊朗德黑兰
elham.cheshmikhani@sharif.edu farbeh@aut.ac.ir

3rd HosseinAsadi计算机工程系 谢里夫理工大学伊朗德黑兰 asadi@sharif.edu

arXiv:2601.00450v1 [cs.AR] 1 Jan 2026

摘要—*Spin-Transfer Torque Magnetic RAM*（**自旋转移力矩磁阻随机存取存储器**）**作为静态随机存取存储器在片上高速缓存中的最有前景的替代方案之一，具有更高的密度和可扩展性、近乎为零的漏电功耗以及非易失性，但其可靠性受到较高读扰动错误率的威胁。***Error-Correcting Codes***（**纠错码**）**通常被用于解决STT-MRAM缓存中的读扰动错误。通过采用高强度纠错码，并在每次读访问时检查缓存块，可以实现较高的缓存可靠性。然而，为了最大限度地缩短现代处理器中的缓存访问时间，目标缓存组中的所有缓存块会同时并行读取，以执行标记比较操作；随后，在检查其纠错码后，仅将请求的缓存块（如果存在）发送出去。这些额外读取的缓存块在处理器请求这些块之前并未检查其纠错码，从而导致读扰动错误累积，显著降低缓存可靠性。本文首先介绍并阐述读扰动累积现象，并指出，由缓存块的常规并行访问导致的这种累积会显著提高缓存错误率。随后，我们提出一种简单而有效的方案，称为** *Read Error Accumulation Preventer cache*（**REAP-cache**），**可在不损害缓存性能的情况下彻底消除读扰动累积。评估结果表明，所提出的REAP-cache可将缓存** *Mean Time To Failure*（**平均故障间隔时间**）**延长171倍，同时使缓存面积增加不到1%，能耗仅增加2.7%。**

索引词—**缓存、自旋转移力矩磁阻随机存取存储器、读扰动、纠错码 (ECC)、错误率**

I. 引言

Static Random Access Memories（SRAM）一直是片上缓存中普遍采用的存储器技术。然而，随着技术缩微，SRAM面临着诸多挑战，例如高漏电功耗和存储单元不稳定性 [1]。非易失性存储器（NVM）技术的最新发展使*Spin-Transfer Torque Magnetic RAM*（自旋转移力矩磁阻随机存取存储器）成为片上缓存中最具前景的SRAM替代方案。近乎为零的漏电功耗、对辐射诱发错误的免疫能力、更高的密度、更好的可扩展性以及非易失性，是STT-MRAM缓存的主要优势 [2]。除上述优势外，STT-MRAM缓存的读操作极易发生错误。在读操作期间向缓存存储单元施加读电流时，存储单元中的内容可能会意外翻转。这种错误称为读扰动，源于随机开关行为自旋转移力矩磁阻式随机存取存储器单元的行为 [3]。读扰动错误是STT-MRAM缓存不可靠性的主要来源。

缓存块通常使用 *Error-Correcting Codes*（纠错码）[4]进行保护。当收到针对某个缓存块的请求时，ECC解码器逻辑会检查该块是否存在可纠正错误，并输出纠正后的数据。然而，当块中的错误位数超过ECC纠错能力时，ECC解码器便无法纠正该数据块。现代处理器中的缓存读取操作经过优化，以缩短访问时间 [5]–[7], ，这显著提高了读扰动导致不可纠正错误的发生概率。

对于针对k路组相联缓存的读取请求，在进行标签比较操作的同时、确定所请求的数据块之前，目标组中的所有块都会同时被读取 [7]–[9] ，随后检查所请求块的ECC，并输出该块。此时，其余 $k-1$ 块会被丢弃。每次读访问中，这些针对未被请求块的额外 $k-1$ 次 k 推测性读取会提高这些块发生读扰动的概率。我们将这些额外施加的读取称为 *concealed reads*。多次读取多个缓存块而不检查其ECC，会导致读扰动错误累积并产生不可纠正错误。据我们所知，以往研究尚未解决这一严重的可靠性问题。

本文中，1）我们首先考察并阐述隐蔽读取对自旋转移力矩磁阻随机存取存储器缓存可靠性的不利影响，并通过分析说明，读扰动累积会显著提高缓存错误率。随后，2）通过开展全面实验，我们证明读扰动累积会大幅降低自旋转移力矩磁阻随机存取存储器缓存的可靠性。接着，3）我们提出一种名为 *Read Error Accumulation Preventer cache*（REAP-cache）的有效方案，以防止缓存块中的读扰动累积，并彻底消除隐蔽读取对缓存可靠性的不利影响。REAP-cache方案只需对缓存读取路径进行少量修改，即可确保不仅对请求的缓存块进行ECC检查，还对缓存组内所有其他并发读取的缓存块进行ECC检查。

我们使用运行一组来自SPEC CPU2006 基准测试套件 [11]的工作负载的gem5全系统模拟器 [10] 评估 REAP-cache。结果表明，REAP-cache可使自旋转移力矩磁阻随机存取存储器缓存的平均故障间隔时间（MTTF） $Mean\ Time\ To\ Failure$平均提高171倍。该方案不会影响缓存性能，缓存面积增加不到1%，能耗开销为$only$ 2.7%。

本文其余部分安排如下。第二节介绍自旋转移力矩磁阻随机存取存储器的基础知识及其可靠性挑战。第三节讨论本研究的观察结果与研究动机。第四节介绍所提出的 REAP-cache 的详细信息。第五节给出仿真设置和评估结果。最后，第六节总结全文。

II. 自旋转移力矩磁阻随机存取存储器基础

如图 1(a) 所示，自旋转移力矩磁阻式随机存取存储器单元由访问元件和存储元件组成。访问元件是一个NMOS晶体管，用于连接或断开该单元与阵列。另一个元件是磁性存储元件，称为 $Magnetic\ Tunnel\ Junction$（磁性隧道结），用于存储数据。磁性隧道结由三层组成：一层薄的氧化物势垒层，将两个铁磁层隔开。该氧化物势垒层由结晶氧化镁构成，夹在$reference\ layer$ 与 $free\ layer$ 之间；前者具有固定的磁场方向，而后者的磁场方向可通过施加电流 [2]改变。自由层的磁化方向可以与参考层相同或相反。如果自旋极化电流改变自由层自旋的取向，使其与参考层自旋平行或反平行，就会形成低电阻或高电阻，分别表示该单元存储逻辑值“0”或“1”。

对自旋转移力矩磁阻随机存取存储器单元进行写入/读取，是通过施加写入/读取电流（I_{write}/I_{read}）并持续预定的脉冲宽度来完成的。要在自旋转移力矩磁阻随机存取存储器单元中写入“0”，I_{write} 会从位线流向源线，使自由层中的电子自旋方向与参考层中的电子自旋方向一致。在单元中写入“1”则需要电流从源线流向位线，并使电子电荷从自由层流向参考层。参考层具有较强的磁场，会反射自旋方向与自由层相反的电子；这些反射电子使自由层的磁场方向反转，从而写入“1”。

要从一个单元读取数据，I_{read} 必须从位线流向源线，或从源线流向位线，以便使用感测放大器测量磁性隧道结的电阻。这意味着，读取操作是单向操作，且方向与写入“1”或“0”时相同。图 1(b) 展示了自旋转移力矩磁阻随机存取存储器单元的读取操作，此时读电流被调整为与写入“0”时的方向相同。在这种情况下，读取操作期间，当I_{read}流经单元时，单元内容可能会意外地从“1”切换为“0”。这种导致单元内容错误改变的意外位翻转称为 $read\ disturbance$。

自旋转移力矩磁阻随机存取存储器单元中发生读扰动的概率由式 (1) [12], [13]计算。

$$P_{Read-Disturbance} = 1 - exp(\frac{-t_{read}}{\tau} \times exp(\frac{-\Delta(I_{read} - I_{C_0})}{I_{C_0}})) \quad (1)$$

其中，τ 为尝试周期，假设为 1 ns；I_{read} 为读电流，I_{C0} 为向 0°K 写入所需的电流，t_{read}为读脉冲宽度，Δ为热稳定因子。

一些研究试图通过减轻读扰动来提高STT-MRAM单元的可靠性。架构级研究中常用的方法是 $disruptive\ reading\ and\ restoring$ [14]、[15]。在这些研究中，每次读取操作后都会执行恢复操作，以纠正错误位。因此，破坏性读取与恢复方法会增加缓存访问时间以及 $write\ failure$概率，带来另一项可靠性挑战。一项被归类为电路级方法的研究尝试通过缩小访问晶体管的尺寸来降低读扰动率 [16]。这种尺寸调整会降低读电流和写电流，从而降低读扰动率；然而，对这种低电流进行感测又会带来一项名为$false\ read$的可靠性挑战。在 [17], [18] 中，研究人员重新设计了感测放大器，以解决无法准确感测的问题，但这会增大单元面积，并使存储器设计更加复杂。

III. 观察与动机

本节将说明，传统缓存配置会在读访问期间使缓存块中的读扰动累积，从而降低缓存可靠性。随后，我们将对读扰动累积对数据块正确性的影响进行建模。最后，我们评估缓存块中这种累积的行为，以阐明该问题的严重性以及缓解这一问题的必要性。

A. 读扰动累积

对于采用纠错码保护的组相联缓存架构，系统会执行一系列操作来完成读取请求。首先，根据请求地址确定缓存组，并将地址的 tag 部分与该组中所有数据块的标签进行比较，以查找匹配项。若缓存命中，则读取请求的数据块，并在检查其纠错码后将其输出

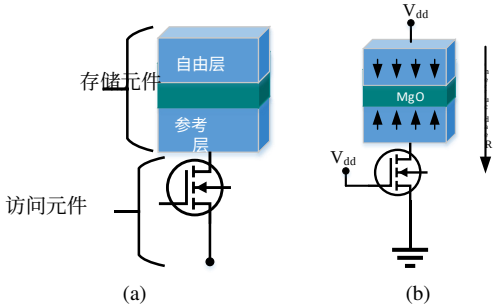

图1：（a）由磁性隧道结单元和访问晶体管构成的自旋转移力矩磁阻随机存取存储器单元结构；（b）通过施加读电流读取自旋转移力矩磁阻随机存取存储器单元。

若缓存未命中，则执行相应的未命中处理流程。为缩短缓存访问时间，传统上会并行执行逻辑上顺序进行的标签比较和从数据块中读取数据这两项操作。

在这种配置下，称为 *cache parallel access* 的 *cache fast access* 模式 [6], [19], [20], ，在标签比较操作完成时，所有数据块都已读取完毕，随时可以发送。缓存命中时，标签比较的输出会从可用数据块中选出一个，其余数据块则被丢弃。这种并行缓存访问省去了访问和读取数据块所需的时间，但由于额外读取数据块，能耗更高。每次访问中，收到针对单个数据块的请求后，都会读取 k 个数据块。

图 2 展示了传统的 k 路组相联缓存，以及缓存快速访问模式下执行的操作序列。在步骤 ，提取输入地址的索引部分，以确定目标缓存组。在步骤 ，读取所选缓存组中的所有标签，并将其与输入地址的标记部分进行比较。同时，对所选缓存组中的所有缓存行进行推测性读取，并将其送至多路复用器单元的输入端，该单元的控制信号来自标签比较单元的输出。在步骤 ，若缓存命中，多路复用器单元会将 k 个数据块中的一个送至 ECC解码器单元，其余 $k-1$ 个数据块则被丢弃；而若缓存未命中，所有 k 个数据块都会被丢弃。缓存命中时选中的数据块会由 ECC解码器单元检查是否存在错误，并在步骤 时发送数据。

如前所述，针对读取单个缓存行的请求，集合中的所有 k 缓存行都会被读取，但只有请求的缓存行中的数据会被检查是否存在错误。其他缓存行虽被读取，却未经过 ECC解码器检查，因此会受到读扰动错误的影响。这些额外读取在本研究中称为 *concealed read* ，会使缓存行中的读扰动错误不断累积，直到该缓存行被请求为止。

在两次纠错码检查之间，一个缓存行可能经历数百次甚至数千次隐藏读取，导致读扰动错误不断累积。如果缓存行中因读扰动错误累积而产生的错误位数超过纠错码的纠错能力，数据就无法恢复。下一节将对以下事件的发生概率进行建模

缓存行中的不可纠正错误。

B. 问题表述

ECC保护的缓存通常能够纠正缓存行中的单比特错误。如前所述，单向读扰动只会影响存储逻辑值“1”的自旋转移力矩磁阻随机存取存储器单元。从某一缓存行读取时，正确数据块传递的概率如式（2）所示。

$$P_{corr-blk} = (1-P_{RD-cell})^{n} + n \times P_{RD-cell} \times (1-P_{RD-cell})^{n-1} \tag{2}$$

其中， $P_{RD-cell}$ 是一次读访问中单元发生读扰动的概率（见式（1）），n是缓存行中“1”的数量。式（2）是一个

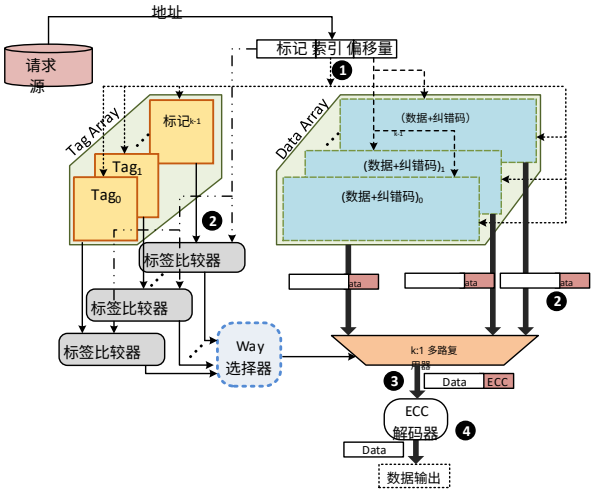

图 2：传统并行访问模式下的缓存结构。

二项试验，其中 $P_{RD-cell}$ 是每次伯努利试验的失败概率，n 是试验次数，而正确数据块传递的概率（$P_{corr_{blk}}$）是失败次数不超过一次的二项概率。

公式 (2) 表明，如果该缓存行中的所有自旋转移力矩磁阻随机存取存储器单元均无错误，或者 n 个单元中只有一个出错，则会传送正确数据。换言之，只要至多有一个数据单元出错，ECC解码器单元就能够传送正确数据 $if\ and\ only\ if$ 。如前所述，在请求缓存行并检查其纠错码之前，该行可能会经历多次隐蔽读取。

每次隐蔽读取中的读扰动概率会在该缓存行中累积，从而降低正确数据传送的概率；该概率根据公式 (3) 计算。

$$P_{corr-blk-acc} = (1-P_{RD-cell})^{N\times n} + N \times n \times P_{RD-cell} \times (1-P_{RD-cell})^{N\times n-1} \tag{3}$$

其中， N 是该缓存行两次连续访问之间发生的隐蔽读取次数加一（用于计入最后一次读访问）。将其视为二项试验，通过比较公式 (2) 和 (3) 可以看出，隐蔽读取会将试验次数从 n 增加到 $N\times n$，而二项概率中的失败次数仍必须至多为一次。

我们通过一个数值示例进一步说明隐蔽读取对不可纠正读操作发生概率的影响。假设一个缓存行的 512 位中有 100 位包含逻辑值“1”，且 $P_{RD-cell} = 10^{-7}$。在没有任何隐蔽读取的情况下，该行经过一次读访问后发生不可纠正错误的概率根据公式 (4) 计算：

$$P_{err} = 1 - P_{corr-blk} = 1 - ((1-10^{-8})^{100} + 100 \times 10^{-8} \times (1-10^{-8})^{99}) = 5.0^{-13} \tag{4}$$

考虑 50 次隐藏读操作时，不可纠正错误的概率根据式 (5) 计算：

$$P\prime_{err-acc} = 1 - ((1-10^{-8})^{100\times 50} + 50 \times 100 \times 10^{-8} \times (1-10^{-8})^{100\times 50-1}) = 1.3 \times 10^{-9} \tag{5}$$

上述示例表明，仅50次隐藏读取就会使无法传递正确数据块的概率增加超过3个数量级。

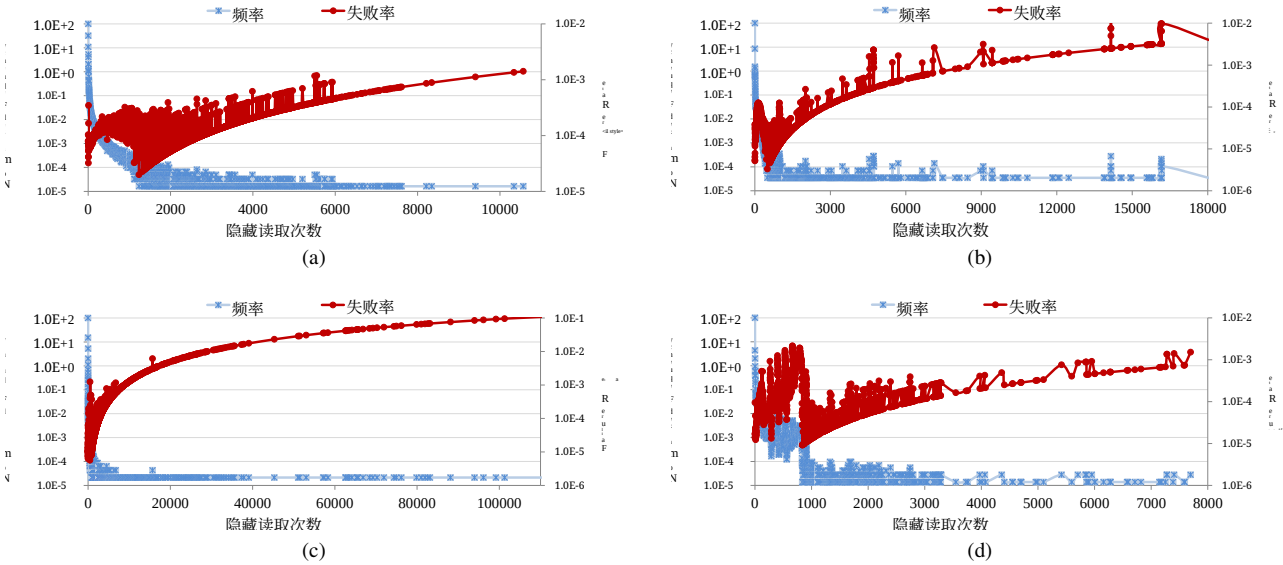

图 3：工作负载执行期间缓存访问中各隐蔽读取次数的频率，以及每种频率对缓存总失效率的贡献：(a) perlbench，(b) calculix，(c) h264ref 和 (d) dealII 工作负载。

上述分析得到了在gem5全系统模拟器 [10], 中对采用 SPEC CPU2006 基准测试套件 [11]工作负载进行评估的8路组相联L2缓存开展的一组实验的支持。评估结果显示，在某些工作负载下，缓存行中的隐藏读取次数甚至可能高于 10^5，这会极大提高多个位发生读扰动的概率。

图 3 展示了四种典型工作负载在执行期间，所有读取访问中不同隐蔽读取次数的频率，以及它们对缓存总失效概率的贡献。具体而言，x 轴表示隐蔽读取次数，主 y 轴表示具有特定隐蔽读取次数的所有缓存行的读取请求数，并将其归一化为没有隐蔽读取的缓存行的读取访问数。例如，点 (35, 3) 表示在工作负载执行期间，发生了 35 次隐蔽读取的缓存行被读取了 3 次；这里将没有任何隐蔽读取的缓存行的读取次数缩放为 100。次 y 轴表示同时考虑隐蔽读取次数及各隐蔽读取次数频率后的失效概率。该概率与频率和隐蔽读取次数均直接相关。主、次 y 轴均采用对数刻度。

结果表明，随着隐藏读取次数增加，每次发生一次隐藏读取的行被读取的频率会降低，而隐藏读取次数较多的行对缓存总失效率的影响则显著增大。例如，在图 3(a) 所示的 $perlbench$ 工作负载中，发生 8000 次隐藏读取的行的读取频率比发生少于五次隐藏读取的行低六个数量级，而前者导致的缓存失效概率比后者高两个数量级。

在 $calculix$ 图 3(b) 中也观察到了相同的趋势，其中具有 10^4 隐蔽读取的缓存行所导致的缓存失效率比隐蔽读取次数少于 20 次的缓存行高出 100 倍以上，尽管前者的频率极低。根据图 3(c)，在$h264ref$ 工作负载中，某些缓存行的隐蔽读取次数超过了 10^5 ，因此它们是缓存总失效率的主要来源。对于 $dealII$ 工作负载，图 3(d) 所示隐蔽读取次数较多的缓存行同样是缓存失效率的主要来源。

上述观察结果表明，为了获得更高的缓存性能，传统缓存配置中的隐藏读取会显著降低自旋转移力矩磁阻随机存取存储器的可靠性。下一节中，我们将提出一种方案，在不影响缓存性能的前提下，消除隐藏读取对可靠性的影响。

IV. 提出的方法

在传统的 k 路组相联缓存结构中，对一个组中某一行的每次读取都会对该组中的其他行额外进行 $k-1$ 次读取。这些隐藏读取会在未检查其纠错码的情况下被丢弃。这会导致读扰动错误累积，并降低读取操作正确完成的概率。为应对这一挑战并消除隐藏读取的不利影响，可以设想两种方法：1）在标签比较操作完成后再读取数据行，但这会失去缓存并行访问带来的性能优势，并显著增加缓存访问时间；2）在一次读访问中对所有数据行执行纠错码检查，而不是只检查请求的行。

我们提出的方案基于第二种方法，可在不影响缓存性能的情况下确保防止读扰动累积。考虑到图 2 所示读取路径上的缓存操作，我们提出了超低成本的 $Read\ Error\ Accumulation\ Preventer\text{-}cache$（REAP-cache）方案，仅对缓存结构进行少量修改。这一修改使得我们能够

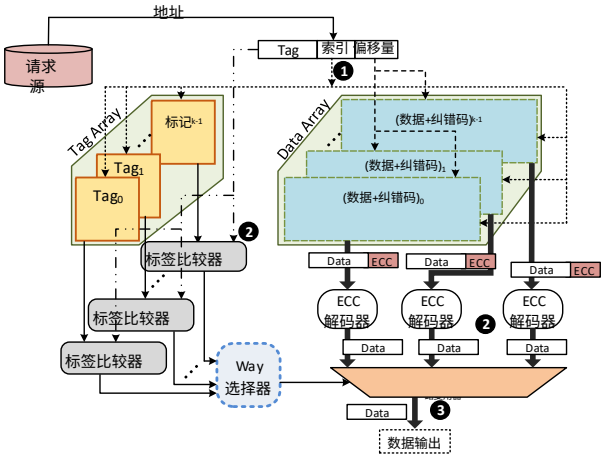

图 4：所提出的 REAP-cache 的缓存结构。

同时检查一次隐蔽读取期间访问的所有数据行的纠错码，以及请求的数据行。

图 4 展示了 REAP-cache 方案中的缓存结构。参照图 2 中的传统缓存结构，在 REAP-cache 方案中，我们交换了读取路径中 ECC 解码器单元与多路复用器单元的位置。为了能够同时检查所有数据块，我们复制 ECC 解码器单元，使其数量与并行访问的缓存行数相同。通过对读取路径进行这一修改，我们保证在每次读取操作后都对所有缓存行执行 ECC 检查，无论读取类型是隐蔽读取还是真正读取。理论上可以证明，交换多路复用器与 ECC 解码器单元的位置不会增加缓存访问时间。

图 4 展示了读取请求中的缓存操作顺序。与传统缓存一样，在步骤 中，根据输入地址的索引部分确定目标组。在步骤 中，读取该组中的所有标签，并将其与输入标签进行比较，同时读取所有数据行。所有 k 数据块都会被发送到 k ECC 解码单元，并在此步骤结束时将纠正后的数据传送到多路复用器单元。标签比较单元的输出会在步骤 中选择一个数据块，随后缓存将该数据发送出去。

REAP 缓存方案通过将每经过 N 次隐蔽读取后检查一次缓存行，改为每次读取操作后检查该行 N 次，从而消除读扰动错误的累积。在这种情况下，该行发生不可纠正错误的概率由式（6）给出：

$$P\prime_{corr-blk-REAP} = ((1-P_{RD-cell})^n + n \times P_{RD-cell} \times (1-P_{RD-cell})^{n-1})^N \quad (6)$$

考虑第三节中之前的示例：一个包含 100 个“1”且经历 50 次隐蔽读取的缓存行，其发生不可纠正错误的概率为 2.6×10^{-11}，比传统缓存低 50 倍。

V. 仿真设置与结果

为评估所提出的 REAP-cache 方案，我们在 gem5 全系统模拟器 [10]中对一款包含两级片上缓存的处理器进行建模。L1 缓存由采用 SRAM 技术的专用数据缓存和指令缓存组成，L2 缓存采用 STT-MRAM 技术，并由多个核心共享。缓存配置详情见表 I。我们使用 SPEC CPU2006 基准测试套件 [11] 作为工作负载，跳过最初的 1 亿条预热指令后，运行 10 亿条指令并提取结果。我们以传统缓存为基准，从可靠性、能耗、面积和性能方面对 REAP-cache 方案进行比较。作为可靠性指标，我们计算 REAP-cache 中缓存的平均故障间隔时间（MTTF），并将其归一化为基准方案的 MTTF。我们使用 NVSim模拟器 [21]对 STT-MRAM 缓存进行建模，以提取能耗、面积和访问时间参数。下文首先研究缓存的可靠性，然后探讨 REAP-cache 方案的开销。

表 I：片上缓存的配置

一级指令缓存	32KB，四路组相联，64字节块大小，写回，静态随机存取存储器
一级数据缓存	32KB，四路组相联，64字节块大小，写回，静态随机存取存储器
L2缓存	1MB，8路组相联，64字节块大小，写回，自旋转移力矩磁阻随机存取存储器

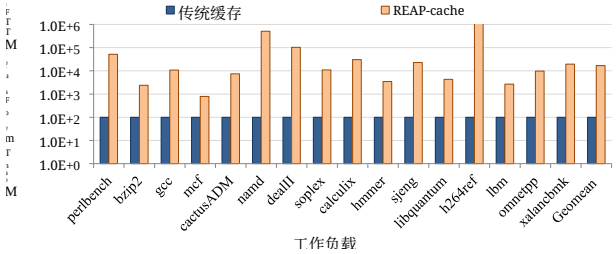

图5：REAP-cache 中缓存的平均故障间隔时间（相对于传统缓存归一化）。

A. 可靠性评估

REAP-cache方案通过消除读扰动错误的累积来提高缓存可靠性。缓存行的每次读访问都可能发生读扰动错误；当该缓存行在多次访问期间未进行检查时，这些错误会累积起来，而当错误数量超过ECC纠错能力时，ECC解码器将无法提供正确的数据。REAP-cache方案在每次对缓存行进行读访问后执行ECC检查，因此ECC解码器只需纠正单次读访问中产生的错误，从而获得更高的纠错能力和更低的缓存失效率。图5展示了在所有工作负载下，REAP-cache方案中自旋转移力矩磁阻随机存取存储器缓存的平均故障间隔时间相对于基准的归一化结果。结果表明，REAP-cache使缓存的平均故障间隔时间提升了171倍。在某些工作负载下，该数值超过1000倍，例如 $namd$、$dealII$、 $h264ref$。在最差情况下，REAP-cache使 mcf 工作负载下的平均故障间隔时间提升了7.9倍。缓存平均故障间隔时间的提升等价于将其错误率平均降低171倍，这得益于REAP-cache提供了更高的纠错能力。

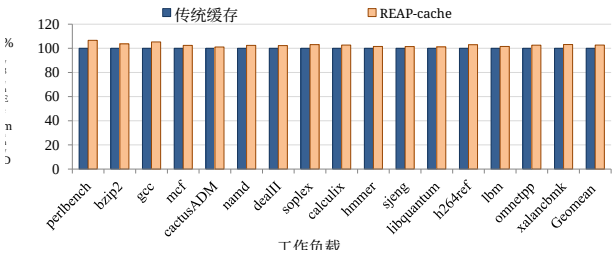

图6：REAP-cache 中动态能耗相对于传统缓存的归一化值。

B. 开销

REAP-cache不会影响缓存写操作中的能耗，也不会影响缓存标记单元操作的能耗。然而，REAP-cache通过在每次读访问中执行更多次ECC检查，增加了缓存能耗。在传统缓存结构中，只有一个ECC解码器单元用于检查请求的缓存行，而REAP-cache则需要八个ECC解码器单元，以同时检查缓存组中的所有数据块。ECC解码器单元在缓存总能耗中的占比低于1%，因此REAP-cache对缓存结构所作的修改对能耗的影响并不显著。

图6展示了 REAP-cache 中 STT-MRAM 缓存相对于基准、针对所有工作负载归一化后的动态能耗。根据结果，REAP-cache 的能耗平均增加了 2.7%。最坏情况下，在 cactusADM 工作负载中观察到 6.5% 的开销；最好情况下，REAP-cache 在 *xalancbmk* 工作负载中产生 1.0% 的开销。能耗开销的差异主要是由于读取访问对总能耗的贡献不同所致。不同工作负载中，读取访问次数占总缓存访问次数的比例以及动态能耗占总能耗的比例各不相同，因此 REAP-cache 的开销会出现小幅变化。

就面积而言，REAP-cache 的开销仅来自引入 ECC 解码器单元。根据我们的评估，ECC 解码器单元占缓存总面积的约 0.1%。因此，在一个 8 路组相联缓存中，将 ECC 解码器单元的数量从一个增加到八个所带来的面积开销不到 1%，可以忽略不计。

从性能角度来看，REAP-cache 不会对缓存访问时间产生不利影响。通过比较图 2 和图 4 中的缓存结构，可以看出，REAP-cache 的读取路径访问时间小于或等于基准缓存。通过交换读取路径中的多路复用器单元和 ECC 解码器单元，REAP-cache 不仅可以让数据行访问重叠，还可以让 ECC 解码操作与标签比较操作重叠。因此，REAP-cache 甚至可以缩短缓存访问时间。

VI. 结论

缓存行与标签比较操作并行访问，以最小化现代处理器中的缓存访问时间。本文揭示了当在片上缓存中以 STT-MRAM 技术替代 SRAM 时，这种优化会导致一个此前未知的可靠性问题，即读扰动累积。在确定所请求的缓存行之前并行访问所有缓存行，会通过提高不可纠正错误的发生概率，显著降低缓存的可靠性。我们提出了REAP-cache方案，以彻底消除读扰动错误的累积。通过对传统缓存结构进行微小修改，REAP-cache使STT-MRAM缓存的平均故障间隔时间平均延长了171倍。这一显著的可靠性提升不会造成性能下降，面积开销低于1%，并且仅使缓存能耗增加2.7%。

参考

[1] M. Tarihi、H. Asadi、A. Haghdoost、M. Arjomand 和 H. Sarbazi- Azad，《利用全面的输入/输出特征分析为固态硬盘设计混合非易失性缓存》，*IEEE Trans. on Comput.*，第65卷，第6期，第1678–1691页，2016年。[2] E. Cheshmikhani、A. M. Hosseini Monazzah、H. Farbeh 和 S. G. Miremadi，《研究工艺变化和系统工作负载对STT-RAM缓存可靠性的影响》，载于*Proc. Euro. Depend. Comput. Conf.*，2016年。[3] T. Na 等，《深亚微米STT-RAM中用于失调消除双级感测电路的读扰动降低技术》，*IEEE Trans. Circ. Syst. II: Exp. Briefs*，第63卷，第6期，第578–582页，2016年。[4] H. Farbeh 等，《Floating-ECC：动态重新定位纠错码位以延长STT-RAM的使用寿命》，*IEEE Trans. Comput.*，第65卷，第12期，第3661–3675页，2016年。[5] H. Farbeh 和 S. G. Miremadi，《Psp-cache：一种低成本容错缓存存储器架构》，载于*Proc. conf. Des., Autom. & Test Euro.* 欧洲设计与自动化协会，2014年，第164页。[6] J. Dai、M. Guan 和 L. Wang，《利用早期标签访问降低嵌入式处理器的L1数据缓存能耗》，*IEEE Trans. on VLSI*，第22卷，第2期，第396–407页，2014年。[7] A. Sembrant、E. Hagersten 和 D. Black-Shaffer，《TLC：一种用于降低动态一级缓存能耗的无标签缓存》，载于*Proc. Symp. Microarchitecture (MICRO)*，2013年，第49–61页。[8] S. K. Mishra 和 V. S. Rao，《在写直达策略下利用路标签信息的高能效L2缓存架构》，2015年。[9] S. Mittal，《提高缓存功耗效率的架构技术综述》，*Sustainable Computing: Informatics and Systems*，第4卷，第1期，第33–43页，2014年。[10] N. Binkert 等，《gem5模拟器》，*ACM SIGARCH Comput. Arch. News*，第39卷，第2期，第1–7页，2011年。[11] J. L. Henning，《SPEC CPU2006基准测试说明》，*ACM SIGARCH Comput. Arch. News*，第34卷，第4期，第1–17页，2006年。[12] Z. Pajouhi 等，《利用故障感知ECC优化STT-MRAM的良率、面积和能耗》，*ACM J. Emerging Tech. Comput. Syst.*，第13卷，第2期，第20页，2017年。[13] A. M. H. Monazzah、H. Farbeh 和 S. G. Miremadi，《LER：新兴STT-RAM缓存的最小错误率替换算法》，*IEEE Trans. Device Mater. Rel.*，第16卷，第2期，第220–226页，2016年。[14] R. Wang、L. Jiang、Y. Zhang、L. Wang 和 J. Yang，《选择性恢复：面向未来STT-MRAM的高能效读扰动缓解方案》，载于*Proc. Des. Automat Conf. (DAC)*，2015年，第1–6页。[15] R. Takemura、T. Kawahara、K. Ono、K. Miura、H. Matsuoka 和 H. Ohno，《面向GB级SPRAM及更大容量的高可扩展破坏性读取方案》，载于*Proc. Memory Workshop (IMW)*。IEEE，2010年，第1–2页。[16] W. Zhao、T. Devolder、Y. Lakys、J.-O. Klein、C. Chappert 和 P. Ma- zoyer，《高可靠性STT-MRAM的设计考量与策略》，*Microelectronics Reliability*，第51卷，第9–11期，第1454–1458页，2011年。[17] T. Na 、 J. P. Kim 、 S. H. Kang 和 S.-O. Jung，《深亚微米 STT-RAM中用于失调消除双级感测电路的读扰动降低技术》，*IEEE Trans. on Circ. Syst. II: Exp. Briefs*，第63卷，第6期，第578–582页，2016年。

[18] W. Zhao、C. Chappert、V. Javerliac 和 J.-P. Noziere，《用于MTJ/CMOS混合逻辑电路的高速、高稳定性、低功耗感测放大器》，*IEEE Trans. on Magnet.*，第45卷，第10期，第3784–3787页，2009年。[19] J. Dai 和 L. Wang，《在写直达策略下利用路标签信息的高能效L2缓存架构》，*IEEE Trans. on VLSI*，第21卷，第1期，第102–112页，2013年。[20] H. Park、S. Yoo 和 S. Lee，《低功耗L2缓存的多步标签比较方法》，*IEEE Trans. on Comput-Aided Des. Integr. Circ. Syst.*，第31卷，第4期，第559–572页，2012年。[21] X. Dong 等，《 NVSim：面向新兴非易失性存储器的电路级性能、能耗和面积模型》，*IEEE Trans. Comput.-Aided Des. Integ. Circuits & Syst.*，第31卷，第7期，第994–1007页，2012年。