



考虑 LDPC 码字的 NAND 闪存信道量化

方泽群, 马征*, 周璇, 庞琦珂

西南交通大学信息科学与技术学院, 信息编码与传输四川省重点实验室, 成都 610031

* 通信作者. E-mail: zma@home.swjtu.edu.cn

收稿日期: 2021-05-06; 修回日期: 2021-07-21; 接受日期: 2021-09-06; 网络出版日期: 2022-08-04

四川省科技计划重点项目 (批准号: 22NSFSC0759) 资助

摘要 随着尺寸工艺的提升和存储方法的演进, NAND 闪存芯片的存储密度正在一步步提高. 而存储密度的提高带来了芯片物理容错性能的下降, 从而导致器件的寿命下降. 如何保证 NAND 闪存的使用寿命是目前闪存器件的研究热点问题. 纠错编码的软判决译码器应用到 NAND 闪存中能够充分提供数据可靠性保障, 但其在数据读取时的量化方法是核心问题之一. 针对目前最大互信息量化方法没有考虑纠错编码码字特性的缺点, 本文提出使用密度进化量化算法来进行基于低密度奇偶校验码 (low density parity check, LDPC) 码字特征的量化. 为了适应 NAND 信道非对称的特点, 本文进一步提出首先利用比特翻转器来将实际非对称译码器输入对称化, 然后通过使用固定比例方法进一步减少搜索最佳读取电压位置的复杂度. 经在 NAND 闪存信道下的数值仿真比较, 本文提出的量化算法可以比最大互信息量化算法更好地提升译码性能, 能够显著提升闪存器件的寿命.

关键词 NAND 闪存, 非对称信道, LDPC 码, 量化, 密度进化

1 引言

在大数据时代浪潮下, 海量数据的存储成为了目前的焦点问题之一. 为了适应市场对存储需求的与日攀升, 全球存储业务正在经历着革命, 其中除了如分布式存储等存储策略的演进, 存储器件也从早期的以磁介质为主导逐渐到了现在的多种介质混合发展. 作为新兴的主流存储器件之一, NAND 闪存凭借其高速随机读写、高可靠性、存储密度高等优良特质极大地刺激了移动业务的发展, 占据大量市场份额的同时也吸引了学术界和工业界的广泛关注.

相比于传统磁盘, NAND 闪存具有强抗震性、抗物理磨损等可靠性方面优势, 但是 NAND 闪存单元中表征数据模式的电荷数量在受到编程擦除循环、单元间干扰、驻留时间损失等因素影响下具有不稳定性^[1~4], 因此仍然需要纠错码来提供数据可靠性保障. 在早期低存储密度的 NAND 闪存产品中,

引用格式: 方泽群, 马征, 周璇, 等. 考虑 LDPC 码字的 NAND 闪存信道量化. 中国科学: 信息科学, 2022, 52: 1542–1559, doi: 10.1360/SSI-2021-0150
Fang Z Q, Ma Z, Zhou X, et al. LDPC code-dependent quantization for the NAND flash channel (in Chinese). Sci Sin Inform, 2022, 52: 1542–1559, doi: 10.1360/SSI-2021-0150

表 1 MLC NAND 6 个月驻留模型下, Code1 和 Code2 量化参数选择对比^[20]Table 1 Comparison between quantitative parameters of the Code1 and Code2 under the MLC NAND 6-month retention model^[20]

Code	<i>R</i> of MMI quantization	<i>R</i> of Monte Carlo simulation
Code1	7	15
Code2	7	7

存储单元中电荷数量的稳定性相对较强, 汉明码、BCH 码等简单硬判决纠错码就可以完成纠错任务. 但是随着制程工艺的发展和存储密度需求的一步步提高, 多级 NAND 闪存中存储单元间隔变小, 数据更容易受到干扰, 超过了简单纠错码方案的保护能力. 在此背景下, 如何在高级 NAND 闪存产品中使用具有更强纠错能力的软判决纠错码方案便成了当前的热点问题, 其中对 LDPC (low density parity check) 码的研究占据了主要地位^[5~9].

LDPC 码是由 Gallager^[10] 于 1960 年提出的一种基于图理论的线性分组码, 该类码的结构可以由稀疏校验矩阵进行描述, 矩阵的结构与译码性能息息相关. 文献 [11, 12] 研究了 LDPC 译码器在不同矩阵结构下的译码性能差异, 文献 [13] 证明了软判决 LDPC 译码器在理想情况下具有接近香农 (Shannon) 限的译码能力. 然而, 虽然 LDPC 码在传统通信系统中已广泛应用, 但其在 NAND 闪存系统中却面临着一些新的挑战^[5]. 例如, 因为器件特性和器件间的电磁耦合效应, NAND 闪存中的数据往往会发生非对称性变化, 这与传统通信系统中常用的对称信道之间存在差异, 而信道差异会直接影响 LDPC 软判决译码器的性能表现.

为了能够在 NAND 闪存中充分发挥软判决译码器的潜能, 如何在实际场景下获取合适的软值输入就成了一个核心问题. 通常比特级软判决译码器的输入与所存储数据为“0”和“1”的概率有关, 而概率值的计算需要 (1) 准确的信号分布估计和 (2) 量化区间设置. 文献 [14, 15] 通过实验以及数据拟合得到合适的函数模型来描述 NAND 闪存信道, 并提出了在线的实时信道估计方案. 文献 [16, 17] 使用了机器学习等算法对存储数据进行了信道概率密度函数估计. 受制于存储设备的读写速率要求, 高精度的量化在日常读写操作中并不实用, 而低精度的量化可以在译码性能和读写速率之间取得平衡. 针对低精度量化门限设置问题, 一种简单的解决方案是采用均匀量化^[18], 然而这一方式在闪存不稳定的信道情况下, 会导致糟糕的译码性能^[19]. 为了提高低精度量化下的译码性能, 最大互信息 (maximum mutual information, MMI) 量化成为了一种被广泛采用的解决办法^[20~22], 文献 [20] 还同时提出在非对称情况下使用固定比例的方法来减少优化变量, 从而降低搜索最优解的复杂度. 此外, 文献 [23] 采用了类似的熵函数计算方法来对等间隔 MMI 量化方法进行改进. 虽然此类基于信道输出概率分布的量化准则为非对称情况下的量化提供了合理的依据, 但是该方法中并没有体现 LDPC 码结构的影响. 例如文献 [20] 中采用了 MMI 量化方法和蒙特卡洛 (Monte Carlo) 仿真量化方法在 MLC NAND 6 个月驻留模型下, 分别对两种码字——Code1 和 Code2 进行了量化参数 *R* (见第 3.2 小节) 的选择对比. 从表 1 可以看到, 在固定信道条件, 使用不同结构码字的情况下, MMI 量化所选择的量化参数间没有区别, 而根据蒙特卡洛仿真结果, 不同码字的最佳量化参数实际上存在明显的差异, 因此使用 MMI 准则进行量化可能会使得 LDPC 码字无法发挥最佳性能 (参考文献 [20] 的图 14 和 15).

因此, 本文在实际非对称 NAND 闪存信道情况下进行了考虑 LDPC 码结构的低精度量化方案的研究, 弥补了当前量化方法所存在的缺陷. 本文贡献总结如下:

- 通过对 MMI 量化算法所存在缺点的研究, 提出考虑 LDPC 码字结构的密度进化算法对 NAND 闪存进行量化, 有效提升了差错保护性能.

- 针对密度进化需要信道输出概率分布满足对称性条件, 提出在 NAND 闪存控制器中使用比特翻转器来使得原本非对称的信道输出概率分布对称化.
- 将固定比例搜索方案与使用比特翻转器的密度进化量化方案相结合, 降低了最佳量化区间的搜索复杂度.

2 系统模型

NAND 闪存存储系统可以看作是一个简化的通信系统, 写入读出操作可以等效为调制解调过程, 存储的数据在存储过程中会受到干扰和噪声的影响, 而应用纠错码可以大幅提升数据保护的可靠性. 本节首先介绍了 NAND 闪存读写操作以及读取电压设置原理, 然后对闪存信道进行了建模并阐述了软判决译码器在 NAND 闪存系统中的应用方式.

2.1 NAND 闪存读写原理

NAND 闪存的最小存储单位为存储单元 (cell), 根据每个单元中存储的比特数目, NAND 闪存可以分为 SLC (single-level cell) 型, MLC (multi-level cell) 型和 TLC (triple-level cell) 型等. 闪存通过字线和位线的共同控制完成读写操作, 每条字线上的存储单元依据单元内所存储的比特数目共同组成了对应的若干页 (page), 一整个存储单元阵列又组成了一个块 (block). 对于目前市面上封装后的闪存芯片来说, 读写操作都是以页为单位的, 一般为若干 kB^[18]. 因为块中的所有存储单元都连接同一根地线, 且存储单元的数据擦除是通过放电来完成的, 所以擦除操作以块为单位进行.

闪存单元中的电荷数目与存储的数据模式相关. 在最大电荷数目的限制下, 数据模式数目的增加会对单元中电荷数目的控制精度提出更高的要求. 在写操作时, 控制器根据待写入的数据模式将对应数量的电荷充入存储单元. 随着存储单元尺寸的不断下降, 单元中所能保存的电荷数逐渐减少. 在面对同样大小的干扰时, 单元内电荷数量越少, 其存储的数据越容易出错. 为了减小数据写入时的单元间干扰, 在目前 1Xnm 制程的闪存芯片中大多采用增量步进脉冲编程 (incremental step pulse program, ISPP) 方式, 即将一次较大的写入电压分为多次较小的写入电压, 将电荷逐级写入目标单元中. 图 1 展示的是 MLC NAND 闪存的 ISPP 示意图, 4 种写入符号分别由 4 种目标状态 (ER, P1, P2 和 P3 状态) 表示, 通过最低有效位 (least significant bit, LSB) 页和最高有效位 (most significant bit, MSB) 页的分别写入完成数据存储功能.

在读操作时, 为了测量存储单元中的电荷数, 目前普遍采用的方式是在存储单元上施加不同强度的读取参考电压, 并通过感应放大器所侦测到的阈值电压大小对电荷数进行定量. 例如要对图 1 中的 MLC NAND 闪存进行硬判决读取, 就将图中垂直黑色实线所示的 3 种不同幅值的硬判决读取参考电压 (r_2 , r_5 和 r_8) 施加于目标单元, 通过感应放大器的导通判断即可以确定该单元所表示的电压值在哪个电压区间中, 从而通过硬判决解映射得到需要的数据.

2.2 存储单元损耗模型

在 NAND 闪存产品中, 时常会出现所存储的数据出现错误或者无法读取的情况, 这极可能是由闪存单元中所存储的电荷量的改变造成的. 文献 [3, 4, 18] 通过实验将改变电荷量的因素总结为驻留时间、编程/擦除循环、邻近单元间干扰、导通电压干扰等若干类. 在这些因素干扰下, 不同电荷数目的单元会呈现不同的反应, 从而造成 NAND 闪存信道的非对称性, 其中一个明显的现象就是图 1 中编程结束后所展示的情况, 处于较低电位 ER 状态的信息表现出比其他 3 个较高电位状态的信息更大的

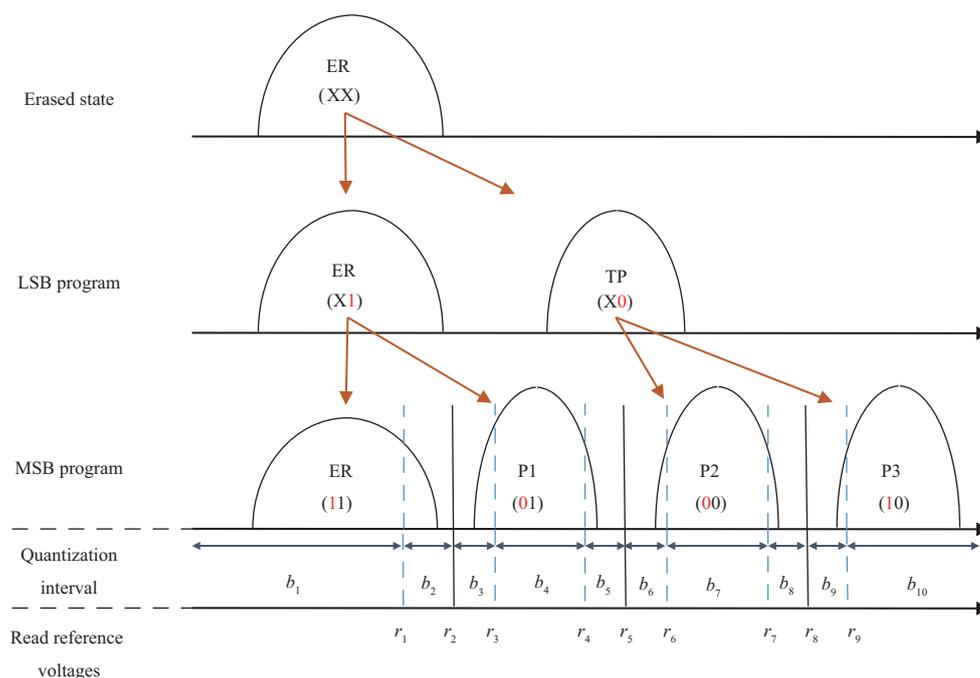


图 1 (网络版彩图) MLC NAND 闪存增量步进脉冲编程过程和量化示意图

Figure 1 (Color online) ISPP and quantization of MLC NAND flash

不稳定性. 为解决不对称信道的建模问题, 文献 [4, 14] 等使用了混合高斯 (Gauss) 模型来分别描述各个状态下信息的分布, 即首先假定各编程状态下的信息分布可以由独立的高斯函数表示, 再通过实验总结不同因素对各个高斯函数中均值和方差的影响, 从而得到一个有效的经验模型. 该方法的数学描述如下:

$$f(y) = \frac{1}{N_s} \sum_{i=1}^{N_s} f_i(y|\mu_i(v), \sigma_i(v)), \quad (1)$$

其中 $f(y)$ 为整体的信道函数, N_s 为编程状态数, $f_i(y)$ 是第 i 个状态所对应的高斯函数, $\mu_i(v)$ 和 $\sigma_i(v)$ 分别是以 v 为变量的各个高斯函数的均值和标准差. 文献 [4] 对 3D MLC NAND 闪存中驻留时间对可靠性的影响进行了研究, 并且通过实验得到了驻留时间损失信道模型, 该文将式 (1) 中的 $\mu_i(t)$ 和 $\sigma_i(t)$ 函数近似为

$$\begin{cases} \mu_i(t) = A_i^\mu \log(t) + B_i^\mu, \\ \sigma_i(t) = A_i^\sigma \log(t) + B_i^\sigma, \end{cases} \quad (2)$$

其中 A_i^μ , B_i^μ , A_i^σ 和 B_i^σ 均为目标条件下的常数, t 是以秒为单位的驻留时间变量.

2.3 译码软信息的获取

相比于硬判决译码器, 软判决译码器具有更强的译码能力, 且在传统通信系统中已被广泛应用. 然而与硬判决译码器直接利用硬判决数据不同, 软判决译码器通常依据接收信息的条件概率进行译码, 结合图 1 中 MLC NAND 闪存的调制方式和式 (1), 以 MSB 页为例, 写入数据 $u = 1$ 和 $u = 0$ 的条件概

率函数分别为

$$\begin{cases} q(y|u=0) = \frac{1}{2} \sum_{i=2,3} f_i(y|\mu_i(v), \sigma_i(v)), \\ q(y|u=1) = \frac{1}{2} \sum_{i=1,4} f_i(y|\mu_i(v), \sigma_i(v)). \end{cases} \quad (3)$$

软判决读取也采用将读取参考电压施加于存储单元进行导通判断的方法来确定存储单元所表示的阈值电压在哪个区间内, 软、硬判决读取的区别在于软判决读取需要在存储单元上多次施加读取参考电压并通过比较器读出软区间下标, 而硬判决读取只需要最少次数 (如 LSB 页需要 1 次, MSB 需要 2 次) 的读取来对信号进行硬判决比较操作. 在存储单元电压所在软区间判断结束之后, NAND 闪存芯片将数据所处的区间下标传输给控制器, 控制器通过映射就可以得到区间所代表的软信息值. 这种映射以对数似然比 (log likelihood ratio, LLR) 软信息表示:

$$L(b_j) = \log \frac{\int_{r_{j-1}}^{r_j} q(y|u=0)dy}{\int_{r_{j-1}}^{r_j} q(y|u=1)dy}, \quad (4)$$

其中 b_j 为量化区间, 如图 1 所示其由读取参考电压 r_{j-1} 和 r_j 确定, j 为 $[1, n]$ 之间的正整数, 且 $r_0 = -\infty, r_n = \infty$. 本文所研究的内容即为读取参考电压 r_j (即量化门限) 的设置问题.

不同于传统通信中模数转换器对接收符号的逐符号串行转换, NAND 闪存中的数据读取以页为单位进行, 而页中的数据量为若干 kB, 这就意味着只有在每个量化门限下完成数以万计的比较器运算之后, 闪存芯片才能将数据传输给闪存控制器. 数据读取过程中的比较操作次数与量化门限数目呈倍数关系, 因此量化门限数目的设置直接影响数据读取的速度, 在实际闪存产品中大部分译码都发生在低精度量化下.

3 NAND 闪存的读取电压量化

相比于使用连续的信道模型, 低精度量化后的离散信道存在一定信息量的损失, 而 MMI 准则可以保证信道容量最大化, 这使得 MMI 成为了目前 NAND 闪存中的一种主流量化准则.

3.1 最大互信息量化

互信息是通信系统中衡量发送端到接收端所传递信息量的一种重要工具. 作为一种新兴的通信场景, NAND 闪存中的数据读写也可以等效为传统通信中信号的发送和接收过程, 因此众多学者将这一工具也引入到了 NAND 存储系统中, 利用 MMI 准则来进行数据处理过程的优化. 式 (5) 为互信息量计算公式,

$$I(X; Y) = H(Y) - H(Y|X), \quad (5)$$

其中 H 为信息熵函数, $H = \sum_i -x_i \log x_i$.

NAND 闪存中的低精度量化多级读取可以等效为图 2 所示的离散无记忆信道. 图 2(a) 和 (b) 表示的是 SLC NAND 闪存中两种软读取情况, 二者区别在于前者通过 2 级量化将信号空间分为了 3 个部分, 其中图 2(a) 中不可靠区间 b_2 的 LLR 值被量化为擦除状态 e ($e = 0$), 而后者使用 3 级量化提供了更高的量化精度.

文献 [20] 给出了对称信道下图 2(a) 和 (b) 中模型的互信息计算公式. 将图 2 中所示的转移概率记作 p_{ij} , i 和 j 分别表示的是信源下标和量化区间下标, 例如 $p_{12} = p(b_2|u_1)$, 该文给出在非对称信道

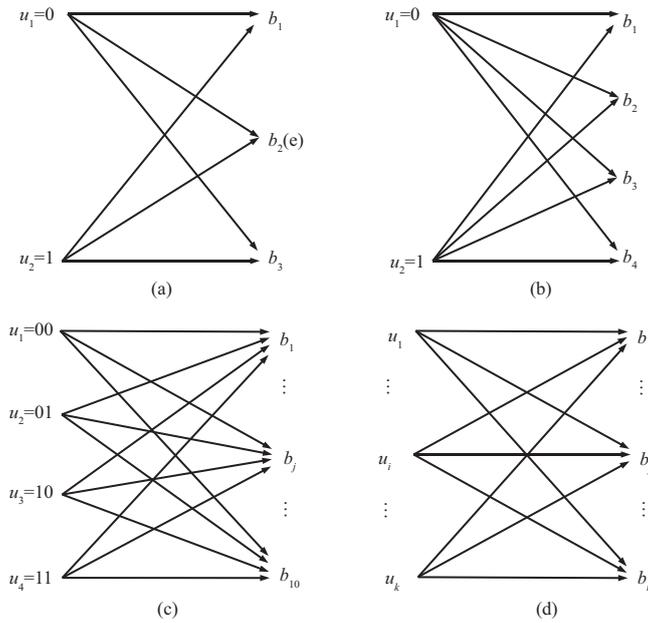


图 2 与 NAND 闪存多级读取等效的离散无记忆信道模型

Figure 2 Equivalent discrete memoryless channel model for multi-reads in MLC NAND flash memory. (a) SLC NAND, 2 reads; (b) SLC NAND, 3 reads; (c) MLC NAND, 9 reads; (d) general model

下该两种模型的互信息量计算公式如下:

$$I_a(X; Y) = H\left(\frac{p_{11} + p_{21}}{2}, \frac{p_{12} + p_{22}}{2}, \frac{p_{13} + p_{23}}{2}\right) - \frac{1}{2}H(p_{11}, p_{12}, p_{13}, p_{21}, p_{22}, p_{23}), \tag{6}$$

$$I_b(X; Y) = H\left(\frac{p_{11} + p_{21}}{2}, \frac{p_{12} + p_{22}}{2}, \frac{p_{13} + p_{23}}{2}, \frac{p_{14} + p_{24}}{2}\right) - \frac{1}{2}H(p_{11}, p_{12}, p_{13}, p_{14}, p_{21}, p_{22}, p_{23}, p_{24}), \tag{7}$$

除此之外, 图 2(c) 和 (d) 给出了 MLC NAND 闪存和更高密度闪存的通用离散量化模型, 同时式 (8) 给出了等概信源、非对称信道情况下的通用互信息计算公式:

$$I_d(X; Y) = \frac{1}{k} \sum_{i=1}^k \sum_{j=i}^n p_{ij} \log \frac{kp_{ij}}{\sum_{i=1}^k p_{ij}}. \tag{8}$$

基于 MMI 准则的量化方案通过搜索求得使互信息量最大的量化门限组合, 将量化问题转化为一个求 $\max I(X; Y)$ 的优化问题, 该方法已被证明具有较好的适用性. 然而值得注意的是, MMI 量化方法只与信道输入输出的概率分布有关, 并没有考虑纠错编译码的影响, 因此, 在考虑编码的情况下, MMI 准则实际上并没有通过联合编译码器与信道对量化门限进行优化^[20]. 本文将于第 4 节提出基于 LDPC 码结构特点的量化方案, 通过联合考虑信道和 LDPC 码字结构完成量化, 弥补 MMI 量化存在的缺点.

3.2 非对称信道量化搜索

在使用 MMI 量化方案时, 需要对优化问题进行求解, 如果所有的量化门限相互独立, 那么多变量求解的复杂度随着存储密度的增加指数上升. 为了降低搜索复杂度, 文献 [19] 通过固定电压量化间隔降低搜索复杂度, 将多变量优化问题转化为了单变量优化问题, 并在对称信道下取得了较好的性能. 但是实际 NAND 闪存信道具有非对称性, 固定量化间隔的方法并非最优选择. 文献 [19] 提出在 MMI 量化时采用一种固定比例 (constant ratio) 的量化方案, 即在相邻两个状态概率分布均值之间的临界区域, 采用式 (9) 所示固定比例的办法来取代固定电压量化间隔, 该方法也被证明是有效的非对称量化简化搜索方法.

$$R = \frac{p(r^l|s_l)}{p(r^l|s_r)} = \frac{p(r^r|s_r)}{p(r^r|s_l)}, \quad (9)$$

其中 r^l 表示当前硬判决门限左边的门限, r^r 表示硬判决门限右边的门限, s_l 和 s_r 分别表示当前硬判决门限左右两边两个编程状态. 值得注意的是, 文献 [23] 采用自定义的熵函数也对固定等间隔 MMI 量化方法进行了优化, 但是该熵函数方法与固定比例的量化门限搜索方法相互等价 (见附录 A), 且基于这两种方法的最佳取值在各自研究中都只能使用蒙特卡洛仿真穷举搜索找到, 而蒙特卡洛仿真的时间并不稳定, 所以这在实际中并不适用.

此外, 该搜索方法在与 MMI 量化方法配合使用时, 仍然无法克服 MMI 算法的缺点. 表 1 中的量化参数选择对比显示在固定信道条件, 使用 Code1 和 Code2 两种不同码字的情况下, MMI 量化所选择的量化参数与最佳量化参数之间会存在明显差异, 因此使用 MMI 准则进行固定比例量化仍然会使得 LDPC 码字无法发挥最佳性能.

4 考虑 LDPC 编译码的 NAND 非对称量化方法

随着 NAND 闪存产品对可靠性需求的不断提高, 软判决译码器凭借相比于传统硬判决译码器更出色的译码性能已逐渐开始获得青睐. 作为当前先进纠错码中的重要一员, LDPC 码不仅具有逼近香农限的性能表现, 还具有内在并行性特点, 这一特点可以使其译码过程以高并行度进行, 从而获得高吞吐率, 具有较大的应用优势 [5, 24]. 本节首先对 LDPC 的译码过程进行了简要概括, 说明码字结构和译码结果之间的关系, 然后提出使用密度进化算法进行联合信道输出和码字结构的量化方法, 同时按照密度进化工具的对称性条件提出利用比特翻转器来保证译信道输出分布的对称性.

4.1 LDPC 译码器

LDPC 码得名于其校验矩阵具有低密度的特点, 矩阵的每一行确定了一组单奇偶校验关系, 行向的非零节点被称作校验节点; 矩阵的每一列构成了一组重复码, 列向的非零节点被称作变量节点. 节点数目反应了矩阵中奇偶码和重复码的连接情况, 极大影响着译码更新过程. 根据变量节点和校验节点的数目, LDPC 码被分为了规则码和非规则码两类. 与规则码校验矩阵中单一的变量节点数目 d_v 和校验节点数目 d_c 不同, 非规则码的节点数目并不统一, 变量节点和校验节点需要使用度分布多项式 $\lambda(x)$ 和 $\rho(x)$ 来进行描述. 根据已有文献的研究, LDPC 的译码性能与其校验矩阵结构具有紧密的关系, 即使采用同码长、码率等参数, 不同的校验矩阵结构也会得到较大的译码性能差异 [14].

LDPC 译码器通常使用消息传递方案来进行译码, 通过在变量节点和校验节点之间进行分布式地迭代更新获得全局最优解, 算法 1 展示了传统和积算法译码过程. 和积算法中信道信息 L^{ch} 采用 LLR 的形式输入译码器, 在校验节点和变量节点更新过程中, $j' \in J \setminus j$ 和 $i' \in I \setminus i$ 表示的是在计算当前节点

算法 1 The LDPC sum-product algorithm**Input:** L^{ch} ;1: $L_{j \rightarrow i}^c = L^{\text{ch}}$;2: **while** $I < I_{\text{max}}$ **do**

3: Check node update:

$$L_{i \rightarrow j}^v = 2 \tanh^{-1} \left(\prod_{j' \in J \setminus j} \tanh \left(\frac{1}{2} L_{j' \rightarrow i}^c \right) \right); \quad (10)$$

4: Variable node update:

$$L_{j \rightarrow i}^c = L_j^{\text{ch}} + \sum_{i' \in I \setminus i} L_{i' \rightarrow j}^v; \quad (11)$$

5: LLR total:

$$L_j^t = L_j^{\text{ch}} + \sum_{i' \in I} L_{i' \rightarrow j}^v; \quad (12)$$

6: Hard decision:

$$\hat{u}_j = \begin{cases} 1, & L_j^t < 0, \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

7: Stopping criteria: if $\hat{u}H^t = 0$ or the number of iterations equals the maximum limit, stop; otherwise, continue the iteration;8: **end while****Output:** \hat{u} .

更新信息 $L_{i \rightarrow j}^v$ 和 $L_{j \rightarrow i}^c$ 时, 并不需要与其连接的节点值 $L_{j \rightarrow i}^c$ 和 $L_{i \rightarrow j}^v$ 参与计算, 其中 J 和 I 表示的是与该节点连接的所有节点集合, 节点集合的数目由校验矩阵的度分布决定.

由算法 1 可以看出, LDPC 译码过程不但与译码器输入的 LLR 相关, 也与校验矩阵的度分布有着密切联系, 对于确定的节点更新关系, 译码器输入信息与度分布共同决定了式 (10) 和 (11) 的更新结果. 然而依据式 (8), MMI 量化方法只考虑了信道输入输出的概率分布, 却无法考虑码字度分布的情况, 因此求解出来的读取参考电压并非是对码字最优. 接下来我们将引入密度进化工具将二者相互关联.

4.2 密度进化与译码器输入对称化

密度进化是基于概率论的一种迭代计算工具, 通过在给定信道下使用密度进化寻找表现最好的 LDPC 码字是一种被广泛应用的 LDPC 码字设计方法^[12]. 与一般使用时间消耗不确定的蒙特卡洛方法进行译码器性能仿真不同, 因为计算过程的确定性, 密度进化可以在确定最大计算时间内预测译码器在给定信道下的性能表现.

LDPC 的密度进化按照图 3 所示的树形译码结构推导所得, 节点间的更新过程从上至下进行, 并隐含假设更新过程中不存在让相邻节点产生相互干扰的环. 给定译码器输入 L^{ch} 的分布函数 p_{ch} , 密度进化依照算法 1 的译码过程对 p_{ch} 进行如算法 2 所示的信号概率分布迭代. 根据概率理论, 随机变量之和的概率分布为其分布之间的卷积, 而随机变量函数的概率分布可以通过公式法进行求解^[25].

算法 2 中式 (14) 和 (16) 分别计算的是式 (10) 校验节点更新和式 (11) 变量节点更新之后的信号概率分布, 其中 l 为迭代次数, $*$ 表示函数卷积. 当 $*$ 出现在上标位置则表示对该分布函数进行连续卷积, 例如 p^{*2} 表示 $p * p$. Γ 表示随机向量在经过 $-\ln \tanh(x/2)$ 函数运算后的密度变化, Γ^{-1} 表示其逆变换^[12]. 因为随机变量的函数分布变换结果只与输入变量分布和更新函数有关, 而与码字结构无关,

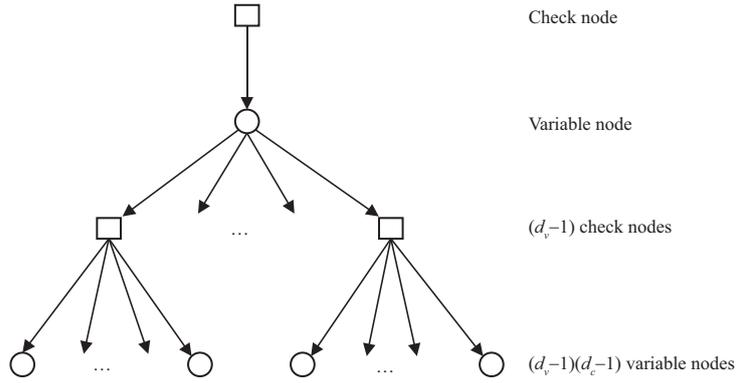


图 3 LDPC 码的树形译码结构
Figure 3 Decoding tree of LDPC codes

算法 2 The density evolution of LDPC codes

Input: p_{ch}, p_e^s ;

1: $p_v^0 = p_{ch}, I = 1$;

2: **while** $I < I_{max}$ **do**

3: Check node update:

$$p_c^l = \Gamma^{-1} \left[\sum_{j=1}^{d_c} \rho_j \left(\Gamma \left[p_v^{l-1} \right] \right)^{*(j-1)} \right]; \quad (14)$$

4: Variable node update:

$$p_v^l = p_{ch} * \sum_{i=1}^{d_v} \lambda_i (p_c^l)^{*(i-1)}; \quad (15)$$

5: LLR total:

$$p_t = p_{ch} * \sum_{i=1}^{d_v} \lambda_i (p_c^l)^{*(i)}; \quad (16)$$

6: Calculation of bit error rate:

$$p_e = \sum_{x=-\infty}^0 p_t(x); \quad (17)$$

7: Stopping criteria: if $p_e \leq p_e^s$, stop; otherwise, continue the iteration;

8: **end while**

Output: p_e .

因此 LDPC 码字的度分布并不参与函数 Γ 与 Γ^{-1} 的计算. 当该 LDPC 码为非规则码字时, d_c 和 d_v 表示的是校验节点度分布和变量节点度分布中的最大节点数, 度分布在密度进化的迭代中决定了节点更新的函数卷积次数. 假设存储的数据为零码字且 LLR 按照式 (4) 进行计算, 演进后预测的错误概率 p_e 为 L_t 小于 0 的概率. 如果给定译码收敛时的预设错误概率 p_e^s , 那么 $p_e \leq p_e^s$ 时迭代终止, 迭代次数代表了译码收敛的速度. 于是在密度进化算法正常工作的情况下, 通过搜索得到迭代次数最小时的最小 p_e , 即可以找到使得译码结果最佳的量化门限组合.

然而, 除了无环假设前提之外, 密度进化工具的使用还需要满足下述对称限制条件.

- 信道输出对称: 信道输出到译码器的信号分布是对称的, 即

$$p_{ch}(y|u = 1) = p_{ch}(-y|u = 0), \quad (18)$$

其中 $p_{\text{ch}}(*)$ 为信道 LLR 的概率分布函数.

- 校验节点对称: 自变量的符号计算可以脱离于幅度计算, 即

$$\Psi_c^{(l)}(s_1 m_1, \dots, s_{d_c-1} m_{d_c-1}) = \left(\prod_{i=1}^{d_c-1} s_i \right) \Psi_c^{(l)}(m_1, \dots, m_{d_c-1}), \quad (19)$$

其中, $s_i \in \{+1, -1\}$.

- 变量节点对称: 自变量符号翻转时的对称性, 即

$$\Psi_v^{(l)}(-m_0, \dots, -m_{d_v-1}) = -\Psi_v^{(l)}(m_0, \dots, m_{d_v-1}), \quad (20)$$

并且 $\Psi_v^{(l)}(-m_0) = -\Psi_v^{(l)}(m_0)$.

在离散非对称 NAND 闪存信道下采用传统解映射方法时, LDPC 译码器的输入为

$$\begin{cases} p'_{\text{ch}}(y|u=1) = \sum_{i=1}^n p_1^i \delta(y - L_q^i), \\ p'_{\text{ch}}(y|u=0) = \sum_{i=1}^n p_0^i \delta(y - L_q^i), \end{cases} \quad (21)$$

其中当 $y = 0$ 时, $\delta(y) = 1$, 当 $y \neq 0$ 时, $\delta(y) = 0$. p_1^i 和 p_0^i 表示的是第 i 个量化区间 b_i 内 $u = 1$ 和 $u = 0$ 的概率, L_q^i 为第 i 个量化区间所对应的 LLR 值, 三者计算公式如下:

$$\begin{cases} p_1^i = \int_{r_{i-1}}^{r_i} q(y|u=1) dy, \\ p_0^i = \int_{r_{i-1}}^{r_i} q(y|u=0) dy, \\ L_q^i = \log(p_0^i/p_1^i). \end{cases} \quad (22)$$

可以看到, 当实际 NAND 闪存信道中比特 0/1 的概率分布不对称时, 信道输出不能满足密度进化的信道输出对称条件, 而在对译码器结构不进行改变的情况下, 传统算法的校验节点和变量节点对称性依然成立, 所以在此情况下使用传统密度进化算法来估计译码器性能的前提是将信道输出概率分布对称化.

定理 1 等概率随机比特翻转/解翻转操作可以将非对称 NAND 闪存信道的错误概率与 LLR 的概率分布同时对称化.

证明 (1) 令非对称 NAND 闪存信道下所存储比特 0 的错误概率为 η_0 , 存储比特 1 的错误概率为 η_1 , 且 $\eta_0 \neq \eta_1$. 当发生等概率随机比特翻转时, 待写入的源数据比特 0 和 1 都有 50% 的概率发生写入状态的改变, 读取到的比特 0 和 1 错误概率分别为 η_0 和 η_1 . 对读取到的数据进行解翻转, 源数据比特 0 的错误概率为 $(\eta_0 + \eta_1)/2$, 源数据比特 1 的错误概率也为 $(\eta_0 + \eta_1)/2$, 因此, 错误概率被对称化.

(2) 令没有随机翻转的写入比特 0/1 的概率分布函数为 p_0 和 p_1 , 当存在等概率随机比特翻转操作时, 未翻转的源比特 0/1 的概率分布函数分别为 $p_{u,0}$ 和 $p_{u,1}$:

$$\begin{cases} p_{u,0} = \frac{1}{2} p_0, \\ p_{u,1} = \frac{1}{2} p_1, \end{cases} \quad (23)$$

源比特 0/1 的翻转后概率分布函数为 $p_{f,0}$ 和 $p_{f,1}$:

$$\begin{cases} p_{f,0} = \frac{1}{2} p_1, \\ p_{f,1} = \frac{1}{2} p_0. \end{cases} \quad (24)$$

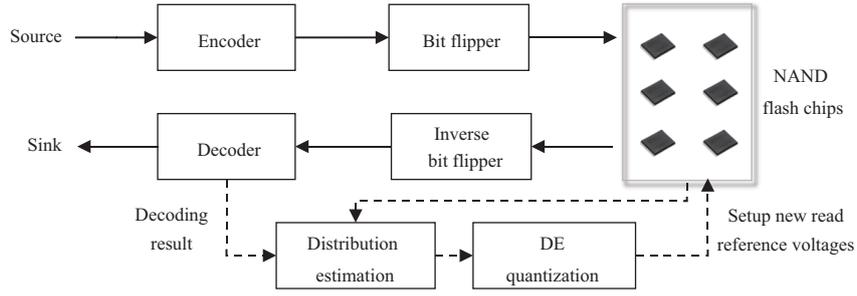


图 4 比特翻转和 DE 量化示意图

Figure 4 Illustration of bit-flipper and DE quantization

于是等概率随机比特翻转/解翻转操作后源比特 0/1 的概率分布 p'_0 和 p'_1 为

$$\begin{cases} p'_0 = \frac{1}{2}p_0 + \frac{1}{2}p_1, \\ p'_1 = \frac{1}{2}p_1 + \frac{1}{2}p_0, \end{cases} \quad (25)$$

其离散形式为

$$\begin{cases} p_0^d = \frac{1}{2}(\sum_{i=1}^n p_0^i + \sum_{i=1}^n p_1^i), \\ p_1^d = \frac{1}{2}(\sum_{i=1}^n p_1^i + \sum_{i=1}^n p_0^i), \end{cases} \quad (26)$$

其中 i 依然表示量化区间的下标.

根据 LLR 的计算公式, 每个读取电压区间内的 LLR 值为比特 1 的概率与比特 0 的概率之比 (式 (4)), 因此根据式 (23), 未翻转比特的 LLR 依然如式 (22). 而根据式 (24), 被翻转比特的 LLR 为

$$L_f^i = \log(p_1^i/p_0^i) = -L_q^i. \quad (27)$$

于是可以得到等概率随机比特翻转/解翻转操作后的离散 LLR 概率分布:

$$\begin{aligned} & \begin{cases} p_{\text{ch}}(y|u=1) = \frac{1}{2} \sum_{i=1}^n p_1^i \delta(y - L_q^i) + \frac{1}{2} \sum_{i=1}^n p_0^i \delta(y - L_f^i), \\ p_{\text{ch}}(y|u=0) = \frac{1}{2} \sum_{i=1}^n p_1^i \delta(y - L_f^i) + \frac{1}{2} \sum_{i=1}^n p_0^i \delta(y - L_q^i), \end{cases} \\ \Rightarrow & \begin{cases} p_{\text{ch}}(y|u=1) = \frac{1}{2} \sum_{i=1}^n p_1^i \delta(y - L_q^i) + \frac{1}{2} \sum_{i=1}^n p_0^i \delta(y + L_q^i), \\ p_{\text{ch}}(y|u=0) = \frac{1}{2} \sum_{i=1}^n p_1^i \delta(y + L_q^i) + \frac{1}{2} \sum_{i=1}^n p_0^i \delta(y - L_q^i). \end{cases} \end{aligned} \quad (28)$$

此时得到 $p_{\text{ch}}(y|u=1) = p_{\text{ch}}(-y|u=0)$, 因此, LLR 概率分布也被对称化.

根据 NAND 闪存一般工作框架与定理 1, 本文在图 4 中提出在闪存控制器中加入比特翻转器来实现译码器输入 LLR 概率分布的对称化, 以保证第 4.3 小节所提量化方法的正常工作.

4.3 量化算法描述

综合上述数据处理方法, 本小节在算法 3 中提出了基于固定比例的非对称信道下密度进化 (density evolution, DE) 量化算法. 根据图 4, 在闪存控制器感知到译码性能发生恶化时, 会对闪存信道进行估计, 得到实时的信道概率密度函数 $f(x)$. 为了应对非对称信道条件, DE 算法首先使用式 (9) 所示固定比例方法按照 R_i 和估计得到的 $f(x)$ 来确定原始信道输出的离散概率分布 p'_i . 然后按照式 (28) 所示比特翻转后的概率分布函数计算公式来将原始信道输出对称化得到 p_{ch} , 将 p_{ch} 作为密度进化算法

的输入就可以计算得到译码器输出错误概率 p_e . 最后, 按照密度进化的门限选取准则, 在迭代结束后再选择使 p_e 最小化的 R_i 来完成读取参考电压的设置.

算法 3 The proposed DE quantization algorithm

Input: The estimated channel model $f(x)$, the degree distribution of the given LDPC code and the initial R_0 ;

- 1: $i = 0$;
- 2: **while** $R_i < R_{\max}$ **do**
- 3: Calculate p'_{ch} according to R_i and $f(x)$;
- 4: Symmetricize p'_{ch} to p_{ch} ;
- 5: Estimate p_e with density evolution;
- 6: $i = i + 1, R_i = R_{i-1} + \Delta R$;
- 7: **end while**

Output: $\arg \min_{R_i} p_e$.

假设闪存控制器已经得到在给定量化参数下的信号分布, 且 MMI 量化方法和 DE 量化方法采用相同的最佳量化参数搜索方法. 在相同量化精度下, 二者的计算复杂度区别在于互信息和密度进化的计算量. 根据式 (8), 互信息的计算主要需要 kn 次加法运算、 kn 次 \log 运算和 $3kn$ 次乘法 (除法) 运算, 因此 MMI 量化的计算复杂度为 $\mathcal{O}(kn)$. 根据算法 2, 密度进化算法的计算量主要由卷积运算构成. 与互信息计算只与符号数 k 和量化数 n 有关不同, 密度进化的复杂度与 (1) 信号分布函数的描述精度有关, 即卷积运算的数据量; (2) 节点更新时的连续卷积运算次数有关, 即度分布情况. 记密度进化中用于信号分布描述的数据量为 n_c , 因为单次卷积运算的复杂度为 $\mathcal{O}(n_c^2)$, 若采用快速傅里叶 (Fourier) 变换, 则其计算复杂度可降为 $\mathcal{O}(n_c \log n_c)$. 记最大卷积次数为 d_{\max} , 密度进化的计算复杂度最低为 $\mathcal{O}(d_{\max} n_c \log n_c)$. 因为通常情况下 $d_{\max} n_c \log n_c > kn$, 因此 DE 量化方法的复杂度高于 MMI 量化方法. 而与文献 [20, 23] 使用蒙特卡洛仿真进行参数选择相比, 因为密度进化计算过程的确定性, 所以 DE 量化方法具有确定计算时间的优势, 利于闪存控制器进行量化门限优化任务的合理安排.

5 数值仿真

本节根据文献 [4] 中总结的 3D MLC NAND 驻留时间损失模型, 使用不同的 LDPC 码字在 MMI 量化和本文提出的 DE 量化方法之间进行数值仿真和验证. 根据 NAND 闪存中高存储密度和长码的需求以及密度进化中的无环假设, 本节首先使用 PEG (progressive edge growth) 算法得到表 2 中所示前两种不同结构 LDPC 码字 [26]: PEG3 和 PEG4, 二者码长分别为 9216 和 8192 bit (≥ 1 kB), 码率分别为 0.89 和 0.9084. PEG 算法使得生成的码字中没有最小环长为 4 的短环, 满足密度进化算法中的无环假设. 此外, 文献 [20] 中码长为 9118 bit, 码率为 0.9021 的码字 1 (Code1) 也在本节中被选用进行仿真对照. 下文译码性能曲线均由蒙特卡洛仿真方法得到, 通过观察在采用不同量化方法 (第 3 节所述 MMI 量化算法和所提 DE 量化算法) 时的译码性能差别, 可以选择相对更合适的量化方法.

图 5 首先展示在无量化情况下利用上述 3 种码字进行有无比特翻转器两种情况下的性能对比. 横坐标 RBER (raw bit error rate) 为硬判决读取后数据的错误概率, 纵坐标 BER (bit error rate) 为译码后的比特错误概率. 从图中可以看到 3 种不同结构的 LDPC 码字虽然具有不同的性能表现, 但是比特翻转器不会对译码性能产生影响.

在比特翻转器的信道输出对称化作用下, 离散信号概率分布在密度进化算法中的演进情况如图 6 所示 (PEG3). 在比特翻转器作用下, 比特 0/1 的错误概率和 LLR 概率分布都具有对称性, 因此图中

表 2 仿真码字
Table 2 Codes for simulation

Code	Length (bit)	Rate	Variable node degree distribution $\lambda(x)$	Check node degree distribution $\rho(x)$
PEG3 [26]	9216	0.89	x^2	$8.1055 \times 10^{-2}x^{25} + 0.8379x^{26} + 8.1055 \times 10^{-2}x^{27}$
PEG4 [26]	8192	0.9084	x^3	$0.3093x^{42} + 0.6907x^{43}$
Code1 [20]	9118	0.9021	$2.0054 \times 10^{-5} + 3.5776 \times 10^{-2}x + 0.39869x^2 + 8.4827 \times 10^{-3}x^8 + 3.7701 \times 10^{-2}x^9 + 0.51933x^{18}$	$0.15662x^{54} + 0.84338x^{55}$

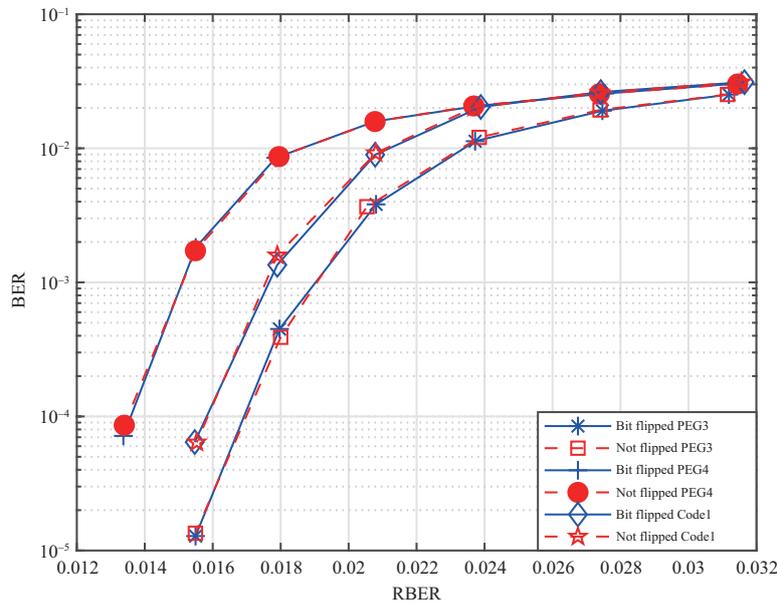


图 5 (网络版彩图) 比特翻转器对译码性能的影响
Figure 5 (Color online) Impact of additional bit-flipper

的密度进化算法使用全 0 码字作为编码器输出 [12], 随着译码器的一次次迭代, 信号的概率分布函数逐渐向 x 轴正方向偏移. 在此情况下, 密度进化算法预测的译码器输出平均错误概率 p_e 就是信号幅度小于 0 的概率之和. 如第 4.3 小节所述, DE 量化准则即是在寻找使得 p_e 最小的量化门限.

文献 [20] 通过仿真验证表明 MMI 量化方法并不能实时得到最佳的 R 值, 并期望通过蒙特卡洛方法得到一个固定的 R 进行译码. 然而图 7 在 6 级量化情况下使用码字 PEG3 来进行不同量化准则下 R 值的对比, 实验发现最佳 R 的取值在不同信道下会发生改变, 而非固定值. 以译码性能的蒙特卡洛仿真结果为基准, 虽然 MMI 量化方法在某些时刻可以与最佳 R 保持一致 (图 7(a)), 但是当信道发生难以预测的改变时, MMI 量化方法的参数选择就会出现明显的偏差 (图 7(b) 和 (c)), 而 DE 量化算法在图 7 中展示了在信道发生变化时可以对 R 值进行实时跟踪的能力. 同时, 图 8 进一步展示了两种量化方法在 R 值选择上的差异性. 从图中可以看到虽然 MMI 量化算法所选择的 R 值在信道改变时也会发生偏移, 但是因为并没有考虑码字结构特点, 在使用不同码字时的结果始终保持一致, 而 DE 量化算法却针对码字结构可以呈现特异性的结果.

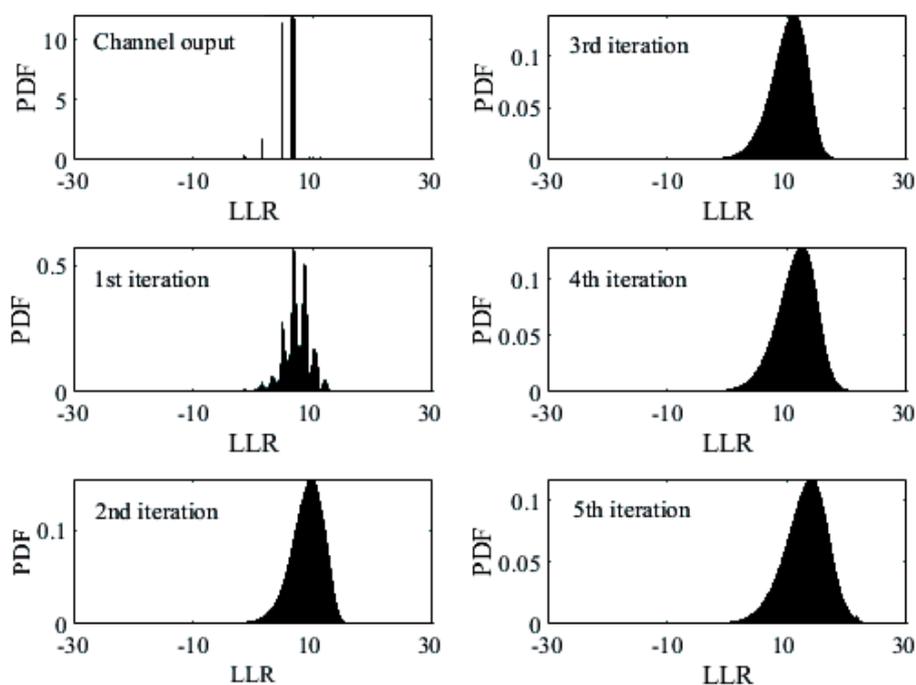


图 6 密度进化过程图例

Figure 6 Illustration of density evolution process

为了验证 DE 量化方法的普适性, 图 9(a)~(c) 在目前常用的 6 级和 9 级量化下^[20, 22, 23], 使用 DE 和 MMI 量化方法对 3 种码字进行了译码性能的蒙特卡洛仿真对比, 其中纵坐标 CER (codeword error rate) 为译码后的码字错误概率. 图中明确显示在不同码字结构下, 本文所提量化算法明显可以提供更好的译码性能. 根据文献 [27], 在差错控制码保护下, RBER 达到最高约 10^{-3} 时存储设备可以被认为不再可靠. 根据文献 [4], 在 10000 P/E 循环时, MSB 页的 RBER 与驻留时间之间具有如下线性关系 (在 MLC NAND 闪存使用寿命后期, MSB 页的 RBER 远大于 LSB 页的 RBER, 所以此处以 MSB 页的 RBER 进行评价):

$$\log \text{RBER} = 0.2149 \log(t) - 11.78. \quad (29)$$

从图 9(c) 中可以观察到 Code1 在 DE 量化方法辅助下可以获得 RBER 约 0.5×10^{-3} 的译码性能增益. 记在采用 MMI 量化方法时, RBER 达到 10^{-3} 时的最大驻留时间为 t_m , 在使用 DE 量化方法时, RBER 达到 0.5×10^{-3} 时的最大驻留时间为 t_d , 根据式 (29) 可得在使用 Code1 时, DE 量化方法可提升数据驻留时间约 $t_d/t_m = 6.6$ 倍. 同理可得, 在 DE 量化辅助下, 使用 PEG3 时 RBER 提升约 0.25×10^{-3} , 驻留时间是采用 MMI 量化方法的 2.8 倍; 使用 PEG4 时 RBER 提升约 0.2×10^{-3} , 驻留时间是采用 MMI 量化方法的 2.3 倍.

经过如上仿真验证, DE 量化算法可以根据 LDPC 码字结构特点选择比 MMI 量化更合适的读取电压门限进行读取数据的量化, 从而最大化纠错编码的纠错能力.

6 总结

本文针对在 NAND 闪存中使用传统 MMI 算法进行量化时量化门限与 LDPC 码字结构不相关的

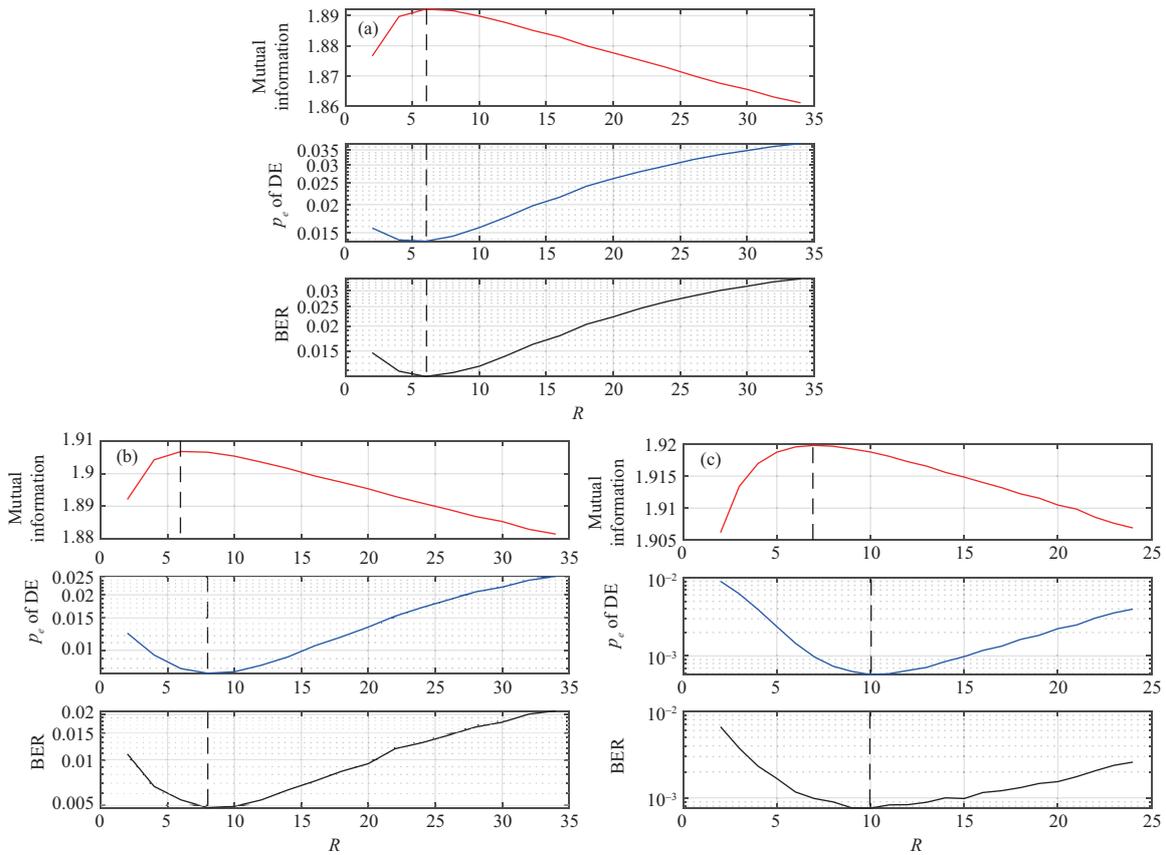


图 7 (网络版彩图) 6 级量化情况下 PEG3 基于 DE 量化、MMI 量化的 R 值选择与蒙特卡洛仿真性能对比

Figure 7 (Color online) Comparison of R values in DE quantization, MMI quantization and Monte Carlo simulation with PEG3 under 6-level quantization. (a) RBER = 1.86×10^{-2} ; (b) RBER = 1.55×10^{-2} ; (c) RBER = 1.29×10^{-2}

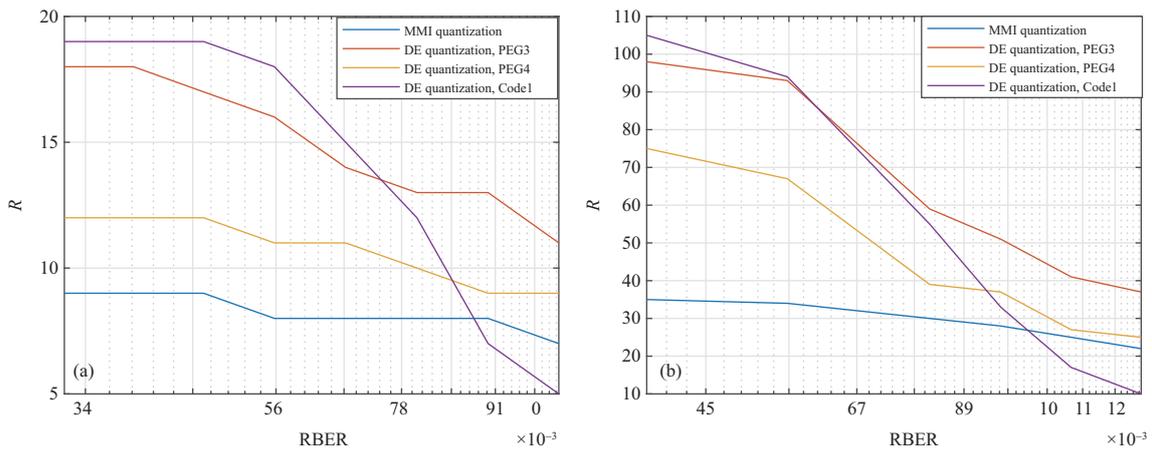


图 8 (网络版彩图) 各码字在不同量化方法下最佳 R 值的选择比较

Figure 8 (Color online) Comparison of the optimal R with different quantization methods and codewords. (a) 6 reads; (b) 9 reads

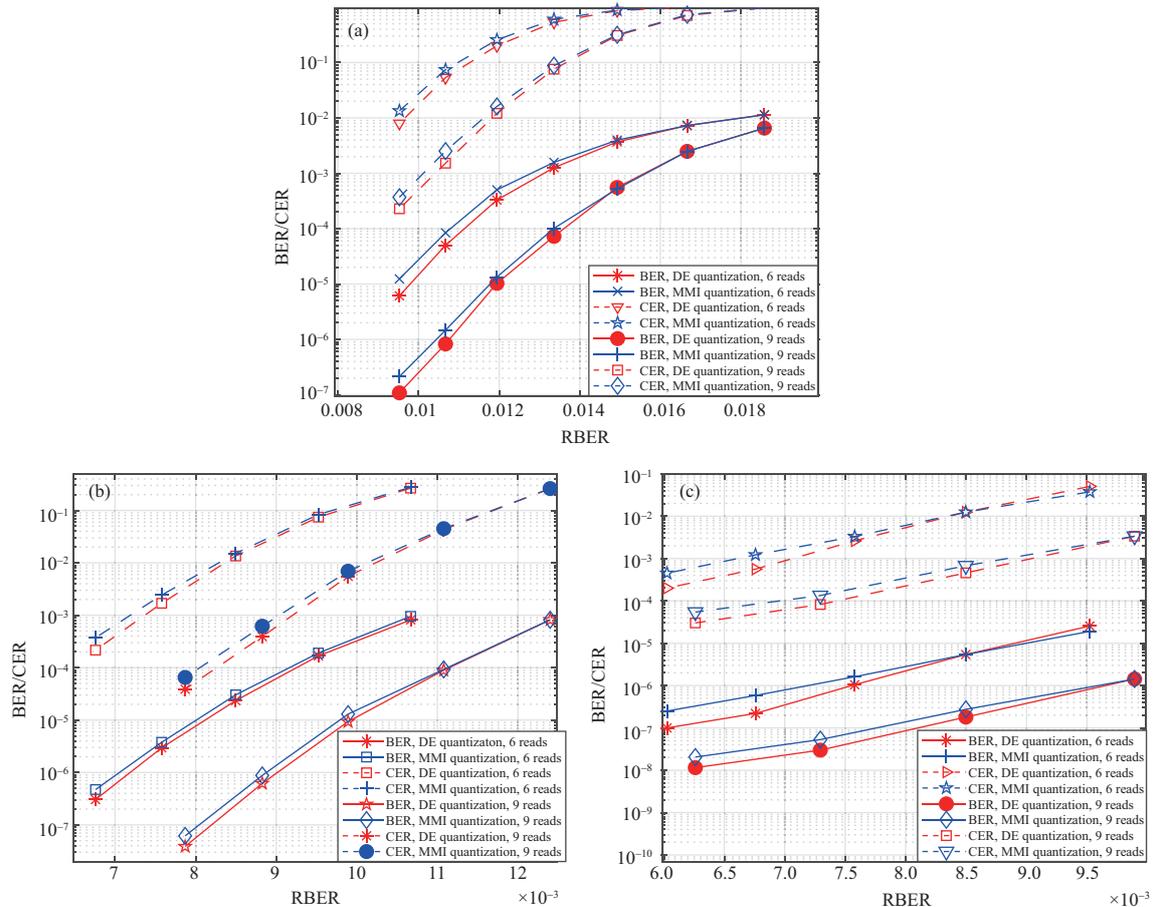


图 9 (网络版彩图) 使用 (a) PEG3, (b) PEG4, (c) Code1 时, MMI 和 DE 量化方法的译码性能对比
Figure 9 (Color online) Decoding performance comparison between MMI and DE quantization with (a) PEG3, (b) PEG4 and (c) Code1

缺点, 提出了一种基于 LDPC 码字的 DE 量化算法. 该算法首先使用固定比例方法来降低非对称信道下量化门限的搜索维度, 将多变量问题转化为了单变量问题进行求解, 接着通过在闪存控制器中设计比特翻转器将非对称译码器输入对称化, 最后利用密度进化算法来进行基于 LDPC 码字的读取参考电压设置. 本文通过使用不同结构的 LDPC 码字在 6 级量化和 9 级量化情况下分别进行了数值仿真和验证, 结果表明所提出的 DE 量化算法可以比传统 MMI 量化算法帮助 LDPC 译码器获得更好的译码性能, 从而达到提高 NAND 闪存可靠度的目的.

参考文献

- 1 Parnell T, Papandreou N, Mittelholzer T, et al. Modelling of the threshold voltage distributions of sub-20 nm NAND flash memory. In: Proceedings of IEEE Global Communications Conference, 2014. 2351–2356
- 2 Papandreou N, Pozidis H, Parnell T, et al. Characterization and analysis of bit errors in 3D TLC NAND flash memory. In: Proceedings of IEEE International Reliability Physics Symposium (IRPS), 2019. 1–6
- 3 Shim Y, Kim M, Chun M, et al. Exploiting process similarity of 3D flash memory for high performance SSDs. In: Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, 2019. 211–223
- 4 Luo Y, Ghose S, Cai Y, et al. Improving 3D NAND flash memory lifetime by tolerating early retention loss and process

- variation. *Proc ACM Meas Anal Comput Syst*, 2018, 2: 1–48
- 5 Dolecek L, Cassuto Y. Channel coding for nonvolatile memory technologies: theoretical advances and practical considerations. *Proc IEEE*, 2017, 105: 1705–1724
 - 6 Oh J, Han S, Ha J. An improved symbol-flipping algorithm for nonbinary LDPC codes and its application to NAND flash memory. *IEEE Trans Magn*, 2019, 55: 1–13
 - 7 Zuolo L, Zambelli C, Marelli A, et al. LDPC soft decoding with improved performance in 1X-2X MLC and TLC NAND flash-based solid state drives. *IEEE Trans Emerg Top Comput*, 2019, 7: 507–515
 - 8 Bu Y, Fang Y, Han G, et al. Design of protograph-LDPC-based BICM-ID for multi-level-cell (MLC) NAND flash memory. *IEEE Commun Lett*, 2019, 23: 1127–1131
 - 9 Liao Y C, Lin C, Chang H C, et al. A (21150, 19050) GC-LDPC decoder for NAND flash applications. *IEEE Trans Circ Syst I*, 2019, 66: 1219–1230
 - 10 Gallager R. Low-density parity-check codes. *IRE Trans Inf Theory*, 1962, 8: 21–28
 - 11 Richardson T J, Urbanke R L. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Trans Inform Theor*, 2001, 47: 599–618
 - 12 Richardson T J, Shokrollahi M A, Urbanke R L. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Trans Inform Theor*, 2001, 47: 619–637
 - 13 Chung S-Y, Forney G D, Richardson T J, et al. On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit. *IEEE Commun Lett*, 2001, 5: 58–60
 - 14 Cai Y, Haratsch E F, Mutlu O, et al. Threshold voltage distribution in MLC NAND flash memory: characterization, analysis, and modeling. In: *Proceedings of Design, Automation Test in Europe Conference Exhibition (DATE)*. Grenoble: IEEE, 2013. 1285–1290
 - 15 Luo Y, Ghose S, Cai Y, et al. Enabling accurate and practical online flash channel modeling for modern MLC NAND flash memory. *IEEE J Sel Areas Commun*, 2016, 34: 2294–2311
 - 16 Peng J, Wang Q, Fu X, et al. Dynamic LLR scheme based on EM algorithm for LDPC decoding in NAND flash memory. *IEICE Electron Express*, 2017, 14: 20170820
 - 17 Tabrizi H, Peleato B, Agarwal R, et al. Improving NAND flash read performance through learning. In: *Proceedings of IEEE International Conference on Communications (ICC)*. London: IEEE, 2015. 370–375
 - 18 Cai Y, Ghose S, Haratsch E F, et al. Error characterization, mitigation, and recovery in flash-memory-based solid-state drives. *Proc IEEE*, 2017, 105: 1666–1704
 - 19 Dong G, Xie N, Zhang T. On the use of soft-decision error-correction codes in nand flash memory. *IEEE Trans Circ Syst I*, 2011, 58: 429–439
 - 20 Wang J, Vakili K, Chen T Y, et al. Enhanced precision through multiple reads for LDPC decoding in flash memories. *IEEE J Sel Areas Commun*, 2014, 32: 880–891
 - 21 Aslam C A, Guan Y L, Cai K. Low-complexity quantization-aware belief-propagation (QA-BP) decoding for MLC NAND flash memory. In: *Proceedings of the 10th International Conference on Information, Communications and Signal Processing (ICICS)*. Singapore: IEEE, 2015. 1–5
 - 22 Wang J, Courtade T, Shankar H, et al. Soft information for LDPC decoding in flash: mutual-information optimized quantization. In: *Proceedings of IEEE Global Communications Conference*, 2011. 1–6
 - 23 Aslam C A, Guan Y L, Cai K. Read and write voltage signal optimization for multi-level-cell (MLC) NAND flash memory. *IEEE Trans Commun*, 2016, 64: 1613–1623
 - 24 Hailes P, Xu L, Maunder R G, et al. A survey of FPGA-based LDPC decoders. *IEEE Commun Surv Tutor*, 2016, 18: 1098–1122
 - 25 Loeve M. *Probability Theory I*. New York: Springer-Verlag, 1977
 - 26 Hu X-Y, Eleftheriou E, Arnold D-M. Progressive edge-growth Tanner graphs. In: *Proceedings of IEEE Global Communications Conference*, 2001. 995–1001
 - 27 Cai Y, Yalcin G, Mutlu O, et al. Neighbor-cell assisted error correction for MLC NAND flash memories. *SIGMETRICS Perform Eval Rev*, 2014, 42: 491–504

附录 A 熵函数方法与固定比例方法的等价性证明

证明 文献 [23] 中所定义的熵函数记为

$$H(r) = \sum_i \frac{p(r|s_i)}{\sum_i p(r|s_i)} \log_2 \frac{\sum_i p(r|s_i)}{p(r|s_i)}. \quad (\text{A1})$$

熵函数方法通过优化硬判决门限左右两边邻近门限的熵值进行设置, 即 $H(r^l) = H(r^r) = \theta$. 只考虑与门限最相关的左右两个相邻状态 s_l 和 s_r , 式 (A1) 可以根据式 (9) 记为

$$H(r) = \frac{p(r|s_l)}{p(r|s_l) + p(r|s_r)} \log_2 \frac{p(r|s_l) + p(r|s_r)}{p(r|s_l)} + \frac{p(r|s_r)}{p(r|s_l) + p(r|s_r)} \log_2 \frac{p(r|s_l) + p(r|s_r)}{p(r|s_r)}. \quad (\text{A2})$$

将式 (9) 中的等式 $p(r^l|s_l) = Rp(r^l|s_r)$ 与 $p(r^r|s_r) = Rp(r^r|s_l)$ 代入式 (A2), 得到

$$H(r^l) = H(r^r) = \log_2(1 + R) - \frac{R}{1 + R} \log_2 R = \theta. \quad (\text{A3})$$

此时, θ 与 R 为单变量关系, 根据文献 [20, 23] 所示最佳门限的唯一性可知, 通过优化 θ 寻找最佳门限和通过优化 R 寻找最佳门限等价, 即熵函数方法与固定比例方法相互等价.

LDPC code-dependent quantization for the NAND flash channel

Zequn FANG, Zheng MA*, Xuan ZHOU & Qike PANG

Key Lab of Information Coding and Transmission, Sichuan Province, School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China

* Corresponding author. E-mail: zma@home.swjtu.edu.cn

Abstract With the improvement of scaling technologies and the evolution of storage strategies, the storage density of NAND flash memory is gradually growing. However, the increase in storage density increases error probability, which reduces the lifespan of NAND flash memory. Error correction coding technology is a powerful tool for ensuring data reliability in NAND flash memory, but how to maximize the error correction ability through quantization in the reading process is an important problem that must be addressed. In view of the shortcomings of the maximum mutual information (MMI) quantization, this study proposes to utilize the density evolution (DE) algorithm to perform quantization on the basis of the features of low-density parity-check codes. To meet the conditions of the DE algorithm, this research also proposes to implement a bit flipper in the NAND controller for symmetrizing the asymmetric input distribution of the decoder. Then, the constant ratio method is chosen to further reduce the complexity of searching for the best read-voltage thresholds. Through numerical simulation and verification, the proposed algorithm is proven to provide better decoding performance than the MMI quantization and significantly improve the lifetime of the NAND flash memory.

Keywords NAND flash memory, asymmetric channel, LDPC codes, quantization, density evolution