2025 Design, Automation & Test in Europe Conference (DATE 2025)

DHD: Double Hard Decision Decoding Scheme for NAND Flash Memory

Lanlan Cui[†‡], Yichuan Wang[†‡], Renzhi Xiao[§], Miao Li[¶], Xiaoxue Liu[†‡] and Xinhong Hei[†‡*]
[†]School of Computer Science and Engineering, XI'AN University of Technology, Xi'an, China
[‡] Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an, China
[§] School of Software Engineering, Jiangxi University of Science and Technology, Nanchang, China
[¶] School of Information Engineering, Zhongnan University of Economics and Law, Wuhan, China
[*]Corresponding author: Xinhong Hei, heixinhong@xaut.edu.cn
{cuilanlan,chuan,liuxiaoxue,heixinhong}@xaut.edu.cn, renzhixiaohust@gmail.com, limiao@zuel.edu.cn

***Abstract*—With the advancement of NAND flash technology, the increased storage density leads to intensified interference, which in turn raises the error rate during data retrieval. To ensure data reliability, low-density parity-check (LDPC) codes are extensively employed for error correction in NAND flash memory. Although LDPC soft decision decoding offers high error correction capability, it comes with a significant latency. Conversely, hard-decision decoding, although faster, lacks sufficient error correction strength. Consequently, flash memory typically initiates with hard-decision decoding and resorts to multiple soft decision decoding upon failure. To minimize decoding latency, this paper proposes a decoding mechanism based on the double hard decision, called DHD. This DHD scheme improves the Log-Likelihood Ratio (LLR) in the hard decision process. After the first hard decision fails, the read reference voltage (RRV) is adjusted to perform the second hard decision decoding. If the second hard decision also fails, soft decision decoding is then employed. Experimental results demonstrate that when the Raw Bit Error Rate (RBER) is 8.5×10^{-3}, DHD reduces the Frame Error Rate (FER) by 86.4% compared to the traditional method.**

***Index Terms*—NAND flash, LDPC, hard decision, LLR**

I. Introduction

As the volume of data continues to grow exponentially, the need for efficient and reliable storage technology has become increasingly pressing. Triple-Level Cell (TLC) NAND flash, a high-density non-volatile storage technology, has gained widespread adoption in consumer electronics, data centers, and cloud computing, among other sectors [1]. Its storage capacity far exceeds that of traditional Single-Level Cell (SLC) and Multi-Level Cell (MLC) flash technology. However, the increasing storage density of TLC NAND flash brings significant challenges in data read reliability, which can adversely affect the overall performance of storage systems and the user experience. To address these challenges, Low-Density Parity-Check (LDPC) codes have been employed in TLC flash memory to enhance data reliability [2], [3].

LDPC codes are implemented through two primary decoding methods: soft decision and hard decision decoding. Soft decision decoding boasts high accuracy but is accompanied by greater computational complexity and time consumption. In contrast, hard decision decoding, while computationally simpler, may exhibit lower decoding accuracy. In practical applications, TLC flash memory systems typically employ a read retry mechanism, starting with hard decision decoding and switching to soft decision decoding if necessary, up to a maximum number of attempts [4]. High Raw Bit Error Rate (RBER) in TLC flash data necessitates multiple re-reads and re-soft decision decoding processes, leading to substantial decoding latency.

In response to this issue, many researchers have contributed to the development of various enhancement techniques. Lv et al. [5] introduced a smart refresh framework to optimize the read reference voltage (RRV), including two refresh schemes: a threshold-based scheme for performance improvement and a periodic scheme for tail latency optimization. Lv et al. [6] proposed a dual error correction code (ECC) method for encoding data with long access latency and low frequency using strong ECC, while employing weak ECC for other data. Liu et al. [7] adjusted the reference voltage for sensing threshold voltage and optimized the initial Log-Likelihood Ratio (LLR) of the iterative LDPC decoder channel. Li et al. [8] proposed asymmetric voltage placement strategies based on asymmetric errors in different states and those caused by voltage shifts. Cui et al. [9] modified the channel initial LLR in both soft and hard decisions to improve LDPC decoding performance.

This paper introduces a novel decoding scheme, termed Double Hard Decision Decoding (DHD), which aims to significantly enhance the overall decoding speed by optimizing hard decision decoding performance and reducing the reliance on soft decision decoding. The core concept of this scheme involves adjusting the RRV to perform hard decoding again after the first hard decision decoding fails, rather than immediately switching to soft decision decoding. This strategy capitalizes on the efficiency of hard decision decoding and maximizes the decoding success rate by optimizing reading parameters. If decoding remains unsuccessful or performance does not meet expectations after these two hard decision decoding attempts, the system will initiate the soft decision iterative decoding process. Experimental results indicate that the DHD scheme exhibits good decoding performance. Specifically, when the

978-3-9826741-0-0/DATE25/© 2025 EDAA

RBER reaches 8.5×10^{-3}, the DHD scheme achieves a reduction in Frame Error Rate (FER) of 86.4% as compared to the conventional approach.

The rest of the paper is organized as follows. In Section II, we provide the background for our work. Section III proposes the DHD. Section IV evaluates the effectiveness of the proposed scheme. Finally, Section V concludes this article.

II. Background

A. NAND Flash Memory and Decoding Scheme

SLC represents the earliest form of NAND flash memory, where each cell is capable of storing a single bit of data. This design ensures high speed and durability but at the cost of reduced storage capacity. Subsequently, the development of MLC and TLC flash extended the storage capacity by allowing each cell to store 2 and 3 bits of data, respectively. However, this increase in density also necessitated trade-offs in terms of speed and durability [10]. TLC NAND flash, in particular, achieves its high storage efficiency by precisely managing the charge within each cell to distinguish among eight voltage states, which are then mapped to three binary digits: the least significant bit (LSB), the central significant bit (CSB), and the most significant bit (MSB) [11], as shown in Fig. 1. This three-digit storage approach significantly enhances storage efficiency in comparison to traditional SLC and MLC flash.

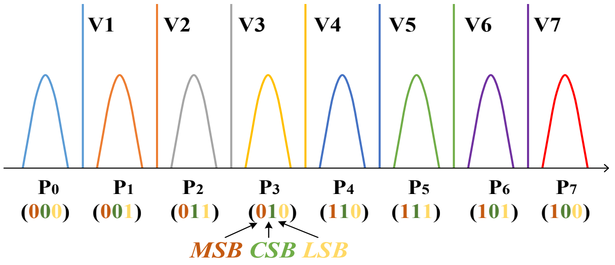

Fig. 1. The eight states of TLC flash.

In the context of flash memory systems, LDPC decoding includes two methods, namely hard decision decoding and soft decision decoding. The hard decision decoding mechanism involves making a direct judgment on the most probable bit value based solely on the received signal, without taking into account the signal's strength or reliability. While this method is straightforward to implement, it may offer limited performance. Conversely, the soft decision decoding mechanism leverages the strength or reliability of the received signal to facilitate more accurate data recovery and thereby enhance decoding performance. This approach, however, is accompanied by a relatively higher computational complexity. When RBER is high, the decoding read retry mechanism is initially employed for TLC NAND flash. First, the hard decision decoding starts by setting an RRV and reading the storage unit according to that threshold. If hard decision decoding fails, indicating a high bit error rate (BER), the system initiates soft decision decoding. Soft decision decoding involves reading the storage unit with multiple reference voltages and calculating the LLR for each bit to obtain the probability information for each bit. If soft decision decoding still fails, the system continues to attempt decoding using soft decision methods until the maximum number of decoding attempts is reached.

B. Threshold Voltage Distribution Shift

The threshold voltage distribution in TLC flash memory is a critical aspect that characterizes the different voltage levels at which a flash cell can store data. Fig. 1 illustrates the threshold voltage distribution of an ideal TLC cell, segmented into distinct data states by a series of RRVs (denoted as $V1, V2, \dots, V7$) [12], [13]. The distribution of threshold voltages in unused TLC flash memory exhibits symmetry. Ideally, during the erasure, programming, and reading processes, the threshold voltages representing different data states in each cell should be confined to non-overlapping voltage ranges to guarantee precise data storage and error-free retrieval. For instance, cells P1 and P2 do not overlap and can be distinctly separated by the voltage $V2$.

In real-world applications, however, the complexity of flash chip fabrication, the rise in storage density, the enhanced interference among cells, and the increase in Program/Erase (P/E) cycles and retention time collectively contribute to a gradual shift in the threshold voltage distribution [14], as depicted in Fig. 2. This shift leads to overlapping and intermingling of the threshold voltage states. If the RRV is not adjusted accordingly, reading errors will escalate. Consequently, it is imperative to adjust the position of the RRV promptly to ensure that it can effectively discriminate between the various data states, thereby significantly reducing the RBER.

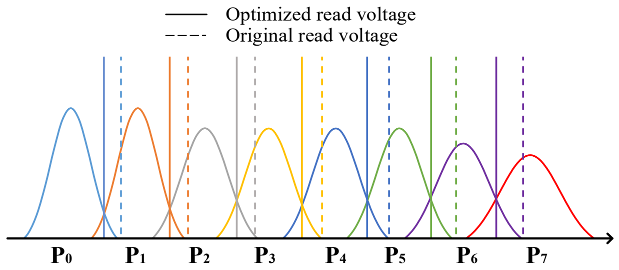

Fig. 2. The threshold voltage distribution shifts.

C. LLR Calculation

In TLC NAND flash memory systems, the calculation of LLR is contingent upon measuring the threshold voltage of each storage cell and comparing it against a predefined RRV to determine the reliability of the bit value (0 or 1) stored within that cell [15]. Assuming we have a function $f(Vth|S_i)$ that denotes the probability density of a flash cell being in state S_i under a specific read voltage Vth, the LLR can be calculated for two consecutive states, S_k and $Sk+1$ (where S_k's threshold voltage is lower than S_{k+1}), as follows:

$$\mathrm{LLR}_{k,k+1}(V_{\mathrm{th}}) = \log \frac{f(V_{\mathrm{th}}|S_{k+1})}{f(V_{\mathrm{th}}|S_k)} \quad .$$

To enhance processing efficiency and minimize storage resource consumption, LLR values are often subjected to quantization. This process involves translating the continuous spectrum of LLR values into a set of predefined, finite discrete LLR value sets, effectively transitioning from a floating-point to a fixed-point representation [16]. These quantized LLR values, which encapsulate key properties of the TLC NAND flash channel, serve as the initial input information for LDPC decoders, facilitating the recovery of the original data during the decoding process.

III. The Proposed Scheme

Our proposed mechanism comprises three integral components, as depicted in Fig. 3. Following the failure of the initial hard decision decoding, the process initiates with the adjustment of the RRV. Subsequently, the data is re-read, and a double hard decision decoding is executed using an optimized channel initial LLR. If decoding still fails, the system will transition to soft decision decoding.

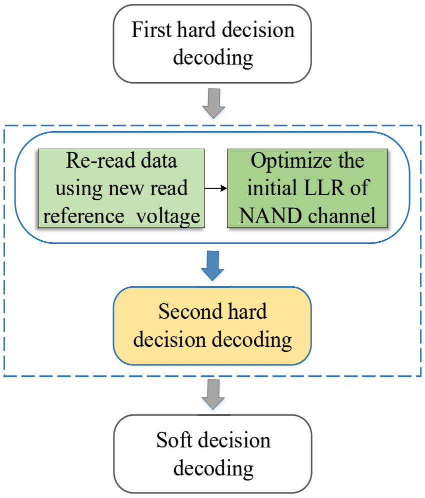
First hard decision decoding
Re-read data using new read reference voltage
Optimize the initial LLR of NAND channel
Second hard decision decoding
Soft decision decoding

Fig. 3. The overview of proposed DHD scheme.

A. Voltage Adjustment

To effectively mitigate the high RBER issue stemming from shifts in the threshold voltage distribution, it is imperative to precisely calibrate the RRV according to these shifts. As depicted in Fig. 4, when threshold voltage shifts occur, the use of a static RRV can lead to high RBER levels, even under improved Eb/N0 conditions during decoding, thereby compromising data transmission reliability. By adjusting the RRV in alignment with the actual shift in the threshold voltage distribution, a marked reduction in RBER can be achieved. Notably, under the Eb/N0 of 1.9dB, the RBER is reduced to an exceptionally low level of 7×10^{-3}.

The determination of the optimal RRV is grounded in the principle of minimizing the area of overlap between the threshold voltage distributions of adjacent storage states. Consider two neighboring states, A and B, with their threshold voltage distributions denoted by functions $f_A(Vth)$ and $f_B(Vth)$. The objective is to identify an RRV that minimizes the area of overlap between the threshold voltage distributions of states A and B.

Calculating the overlapping area: Define two integral intervals: $I1 = (-\infty, RRV]$ and $I2 = (RRV, +\infty)$. The overlapping area S consists of two parts: the area of state B's threshold voltage distribution within interval $I1$ and the area of state A's threshold voltage distribution within interval $I2$. Therefore, S can be represented as:

$$S = \int_{-\infty}^{RRV} f_B(V_{th})\, dV_{th} + \int_{RRV}^{+\infty} f_A(V_{th})\, dV_{th}.$$

Optimal RRV Solution: To determine the RRV that minimizes the overlapping area S, a binary search algorithm is employed. Initially, a practical range for the RRV search is established, e.g., $[V_{th,min}, V_{th,max}]$, where $V_{th,min}$ and $V_{th,max}$ represent the minimum and maximum threshold voltages of all storage states, respectively. An initial RRV value is then chosen within this search range, and the overlapping area $S0$ at this point is calculated. Subsequently, the search range or the RRV value is adjusted based on the comparison between $S0$ and the overlapping area calculated after RRV adjustments, gradually converging towards the optimal solution. During this process, the RRV is typically adjusted towards the direction where the overlapping area is decreasing. The procedure is repeated until a convergence criterion is met, such as the change in the overlapping area falling below a predefined threshold or a predefined number of iterations is reached. At this juncture, the determined RRV value represents the optimal RRV.

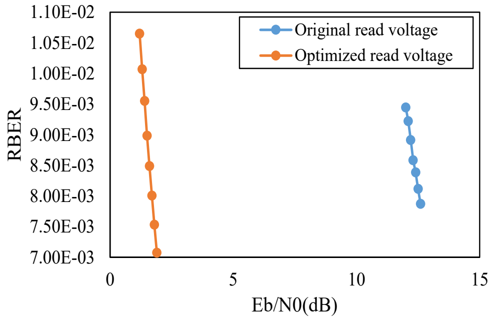

Fig. 4. The RBER comparison of different read voltages used.

B. Optimizing Hard Decision LLR

In TLC flash, the process of data storage and retrieval involves multiple bit positions (LSB, CSB, MSB), each of which, due to their distinct physical properties and positions, experiences and is influenced by interference in varying degrees during reading, leading to different probabilities of misinterpretation during the reading process. To enhance decoding performance and ensure precise data recovery, it is essential to conduct fine-grained processing based on the reliability variances of each bit position. This entails precisely adjusting the LLR values corresponding to each bit position to bolster the accuracy of LLR for the LSB, CSB, and MSB pages, thereby optimizing the overall decoding efficacy.

The data reading process of TLC flash cells is intricate and requires delicate handling, with the distinction between different bit positions achieved through a series of meticulously designed RRVs. Specifically, an RRV $V4$ is first employed to discern the value of the MSB. Then, two RRVs $V2$ and $V6$ are utilized to judge the value of the CSB. Finally, four RRVs $V1, V3, V5$, and $V7$ are employed to finely differentiate the value of the LSB. This multi-layered reading strategy embodies the dual objective of TLC flash: maximizing data storage density while ensuring reading accuracy.

However, it is important to recognize that the reliability of the LSB in TLC flash is comparatively low. As the number of flash read-write cycles increases, the error rates for all three bit pages (LSB, CSB, MSB) escalate, with the LSB experiencing a more pronounced increase in error rate, surpassing that of the CSB and MSB. This discrepancy in reliability underscores the need for heightened attention and optimization of the LSB during the decoding process.

To address the challenge of low LSB reliability, reference [9] suggests an enhanced strategy, which involves reducing the LSB value is set to 13, while maintaining the CSB and MSB values is 16. This adjustment aims to decrease the misjudgment probability during LSB reading by reducing the quantization level, thereby enhancing overall decoding performance. The implementation of this strategy necessitates corresponding adjustments and optimizations to the hardware design and decoding algorithm of TLC NAND flash.

C. Double Hard Decision Decoding

In the decoding process of TLC flash, the read latency of the flash can be divided into three parts: read data time T_{read}, data transfer time $T_{transfer}$, and decoding time T_{decode}. Among them, decoding time is further composed of hard decision decoding time T_{hard} and possibly multiple soft decision decoding times $n \times T_{soft}$, where n is the number of soft decision decoding iterations. Therefore, the formula for decoding latency can be expressed as: $T_{decode} = T_{hard} + n \times T_{soft}$.

To reduce the overall read latency, considering that the data transfer rate $T_{transfer}$ and read data time T_{read} are usually limited by hardware characteristics and difficult to optimize directly, this paper focuses on reducing decoding latency. The specific strategies include:

1. Dynamically adjusting the RRV. In the case of hard decision decoding failure, dynamically adjust the RRV and re-read the data.

2. Optimizing the hard decision decoding LLR. By improving the hard decision decoding LLR, the BER after decoding can be reduced. This improvement can directly postpone the triggering of soft decision decoding, as soft decision decoding is typically triggered only after hard decision decoding fails. Therefore, reducing the error rate of hard decision decoding means reducing n, thereby reducing the overall decoding latency.

3. Double hard decision decoding. Use the optimized hard decision decoding LLR for extra hard decision. This strategy utilizes the statistical characteristics of the threshold voltage distribution of flash cells, optimizing the performance of hard decision decoding by fine-tuning the RRV, which may avoid unnecessary soft decision decoding processes, further reducing n and the overall read latency.

The execution flow of DHD is presented in Algorithm 1. Firstly, parameters are read into the params variable. If the initial read fails (determined by $FailedToRead(params)$), the reference voltage is adjusted to V_{new}, and an attempt is made to reread the data. If this read is successful, hard decision decoding is performed based on these parameters, and a check is made to see if the decoding was successful. If hard decision decoding succeeds, the result is output. If it fails but the maximum number of attempts has not been reached, a double hard decision decoding is attempted, and again, a check is made for decoding success. If all attempts at hard decision decoding fail or are not suitable for the current situation, the process moves on to soft decision decoding. During the soft decision decoding phase, a soft decision reference operation is first performed, followed by reading the data into the $softParams$ variable, and an attempt is made to perform soft decision decoding. If soft decision decoding succeeds, the result is output. If it still fails but the maximum number of attempts has not been reached, an additional read retry is continued.

Algorithm 1 Pseudo-code for Data Reading and Decoding

```
Read parameters into $params$
if FailedToRead ($params$) then
    Set reference voltage to $V_{new}$
    Read data with $V_{new}$ into $params$
    if SuccessfulRead ($params$) then
        Perform hard decision decoding on $params$
        if Decoding successful then
            Output result
        else if not max attempts reached then
            Perform secondary hard decision decoding
            if Secondary decoding successful then
                Output result
            else
                Proceed to soft decision
            end if
        end if
    end if
end if
Perform soft decision reference
Read data with soft decision reference voltage into $softParams$
Perform soft decision decoding on $softParams$
if Soft decision decoding successful then
    Output result
else if not max attempts reached then
    Additional retry
end if
```

IV. Experimental Results and Analysis

A. Experimental Setup

In our experiments, we have chosen to employ the min-sum algorithm for decoding purposes, which is a popular choice due to its balance between decoding performance and computational complexity. We utilize a 4K quasi-cyclic low-density parity-check (QC-LDPC) code. QC-LDPC codes are a subclass of LDPC codes that exhibit a quasi-cyclic structure in their parity-check matrix. This structure not only simplifies the encoding and decoding processes but also helps in reducing the hardware complexity for implementation.

Upon the completion of algorithm design, its deployment to real-world hardware systems necessitates consideration of the inherent complexities and resource-intensive nature of floating-point arithmetic at the hardware level. Consequently, the conversion of floating-point numbers to fixed-point representations emerges as a prevalent optimization technique. This conversion necessitates not only a judicious tuning of numerical precision but also an accommodation of the bit-width constraints inherent in the hardware. For instance, when the initial information originating from the flash channel is conveyed to the decoder, and if the hardware architecture imposes a 6-bit bit-width limitation, the maximum amplitude value of this initial information would be $2^{(6-1)} - 1$, equating to 31. Consequently, all LLRs are then scaled to fit within the range of $[-31, 31]$.

In practical implementations, the flash system may uniformly preset the initial amplitude values of the LSB, CSB, and MSB to 31. However, considering that the LSB typically has the lowest reliability, and the LLR is a measure of data reliability, a smaller LLR value should be allocated for the low-reliability LSB. According to reference [9], in our double hard decision mechanism, the LLR amplitude value for the LSB is adjusted to 13, while the LLR values for the CSB and MSB remain at 16. The final LLR table, reflecting these adjusted values, is detailed in Table I.

TABLE I
The LLR of different stages

State	P0	P1	P2	P3	P4	P5	P6	P7
LSB	13	-13	-13	13	13	-13	-13	13
CSB	16	16	-16	-16	-16	-16	16	16
MSB	16	16	16	16	-16	-16	-16	-16

B. Results Analysis

Fig. 5 illustrates the comparison of FER among three different methods. The traditional method refers to the single hard decision approach currently adopted in decoding process. Reference [9] introduces an improvement to the initial LLR of the channel based on a single hard decision decoding. As the RBER increases, the FER of the decoding results also shows a gradual upward trend, demonstrating a positive correlation between RBER and FER. The highest FER is observed in the traditional method, followed by the approach in reference [9]. DHD has the lowest FER. For example, when the RBER is 8.5×10^{-3}, the DHD method reduces the FER by 86.4% compared to the traditional method and by 54.7% compared to reference [9], significantly improving the decoding performance.

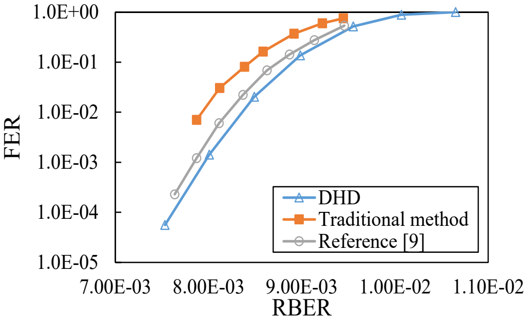

Fig. 5. The comparison of the FER.

Fig. 6 compares the number of iterations required by three methods. As the RBER gradually increases, the average number of iterations needed for the decoding process shows a continuous upward trend. Among these algorithms, the DHD mechanism proposed has the least number of iterations, followed by the approach in reference [9], with the traditional method requiring the most iterations. For example, when the RBER is 8.37×10^{-3}, the DHD requires 9.5 iterations, while the traditional method requires 12.84 iterations. Compared to the traditional method, DHD reduces the number of iterations by 26%. The difference in iterations between DHD and reference [9] is not significant, with DHD decreasing the iterations by 4.3%.

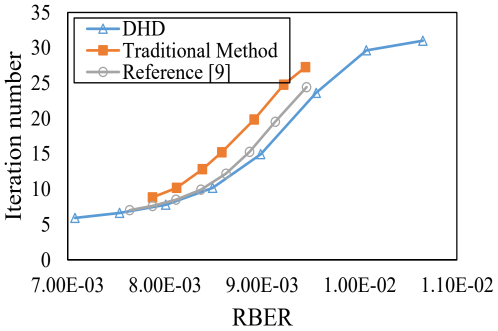

Fig. 6. The comparison of the iteration number.

Decoding latency is positively correlated with the number of iterations. The low number of iterations for DHD means it performs exceptionally well in terms of decoding latency. Compared to traditional methods, DHD significantly reduces the total time required for decoding by minimizing unnecessary iterative cycles. The decoding method proposed in reference [9] also exhibits a lower number of iterations, but its advantage over DHD is not as pronounced. This indicates that while the method in reference [9] shows improvement over traditional methods in terms of decoding latency, it still has a certain gap compared to DHD.

We compare the number of iterations for the three algorithms under different combinations of P/E cycles and retention times, as shown in Fig. 7. To simplify the expression, we refer to the combination of P/E cycles and retention times as (PE, RT). For example, (1,3) indicates 1K P/E cycles and 3 months of retention time. Through the analysis of the experimental data, it can be clearly observed that under all tested (PE, RT) combinations, our proposed DHD strategy consistently exhibited the lowest number of iterations. In comparison, the traditional mechanism and the algorithm proposed in reference [9] had a similar number of iterations, but the algorithm in reference [9] showed slightly fewer iterations in most cases, slightly better than the traditional mechanism. This result indicates that our DHD strategy has a significant advantage in efficiency, being able to effectively reduce the number of iterations and thereby improve the read performance of TLC NAND flash.

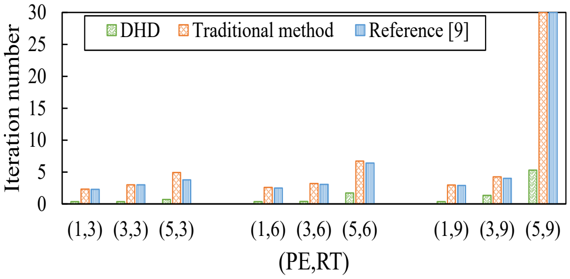

Fig. 7. The comparison of the iteration number for various P/E cycles and retention time.

The proposed DHD method has the following advantages:

1. Reduction of decoding latency. Hard decision decoding is fast, and by adding one more hard decision decoding, the number of soft decision decoding iterations is reduced. Decreasing the number of soft decision decoding iterations can lower the overall decoding latency, thereby reducing the system's read latency.

2. Narrowing the gap between soft and hard decision. Traditional hard decision methods have a certain gap with soft decision decoding. The improved method, during the hard decision process, reduces the gap between soft and hard decision by optimizing the performance of hard decision decoding, making it closer to soft decision decoding. This method can improve decoding accuracy and reduce the bit error rate.

However, this method also has certain limitations: it can only correct errors with an RBER below 1.07×10^{-2}. Beyond this range, error correction can only be achieved through soft decision decoding. Therefore, the applicable range of this method is when RBER $< 1.07 \times 10^{-2}$. This limitation indicates that under high RBER conditions, soft decision decoding remains indispensable.

The DHD scheme proposed in this paper innovatively optimizes the traditional one-step hard decision decoding process into a double hard decision decoding process. In the traditional method, one hard decision decoding is usually followed by eight soft decision decoding steps to enhance decoding accuracy. In contrast, the DHD mechanism adopts two steps of hard decision decoding, followed by seven steps of soft decision decoding. The core of this adjustment is that when the first hard decision decoding fails, the DHD mechanism does not directly switch to soft decision decoding but instead chooses to re-execute the hard decision decoding. This strategy fully utilizes the fast processing speed of hard decision decoding. It is noteworthy that both the soft decision and hard decision required LLR values are pre-calculated and stored in the LLR table, so this change almost does not increase additional storage overhead. At the same time, since the process of re-reading data is the same under both mechanisms, the time overhead also remains unchanged. However, due to the fast processing speed of hard decision decoding itself, the DHD mechanism effectively reduces the overall decoding latency by reducing the number of soft decision steps and adding one hard decision step, thereby improving decoding efficiency.

V. Conclusion

This paper introduces a novel double hard decision decoding mechanism, referred to as DHD. Following the failure of the initial hard decision decoding, DHD modifies the voltage levels to re-read the data and optimizes the initial LLR values conveyed from the TLC NAND flash channel to the LDPC decoder. Thereafter, a second round of hard decision decoding is executed utilizing these enhanced LLR values. If the second hard decision decoding fails, perform soft decision decoding. Experimental findings reveal that DHD notably decreases the number of decoding iterations and substantially reduces the FER. Notably, when the RBER reaches 8.5×10^{-3}, DHD achieves an 86.4% reduction in FER compared to the conventional method.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. U2468206 and No. U2468203.

References

[1] D. Wei, Y. Wang, H. Feng, H. Xiang, and L. Qiao, "LLD: Lightweight latency decrease scheme of LDPC hard decision decoding for 3-D TLC NAND flash memory," *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 1–13, 2024.

[2] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[3] M. Zhang, X. Zhang, F. Wu, K. Tao, F. Zhu, S. Li, Y. Zhao, and C. Xie, "ALCod: Adaptive LDPC coding for 3-D NAND flash memory using inter-layer RBER variation," *IEEE Transactions on Consumer Electronics*, vol. 69, no. 4, pp. 1068–1081, 2023.

[4] J. Cui, Z. Zeng, J. Huang, W. Yuan, and L. T. Yang, "Improving 3-D NAND SSD read performance by parallelizing read-retry," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 3, pp. 768–780, 2023.

[5] Y. Lv, L. Shi, Q. Li, C. Gao, C. J. Xue, and E. Sha, "Optimizing tail latency of LDPC based flash memory storage systems via smart refresh," in *2019 IEEE International Conference on Networking, Architecture and Storage (NAS)*, 2019, pp. 1–8.

[6] Y. Lv, L. Shi, L. Luo, C. Li, C. J. Xue, and E. H.-M. Sha, “Tail latency optimization for LDPC-based high-density and low-cost flash memory devices,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 3, pp. 544–557, 2022.

[7] W. Liu, G. Han, R. He, Y. Fang, and G. Cai, “Dynamic-reference-voltage-based detection algorithm for LDPC-Coded NAND flash memory,” in *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2018, pp. 1–5.

[8] Q. Li, L. Shi, C. J. Xue, Q. Zhuge, and E. H.-M. Sha, “Improving LDPC performance via asymmetric sensing level placement on flash memory,” in *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2017, pp. 560–565.

[9] L. Cui, F. Wu, X. Liu, M. Zhang, R. Xiao, and C. Xie, “Improving ldpc decoding performance for 3D TLC NAND flash by LLR optimization scheme for hard and soft decision,” *ACM Trans. Design Autom. Electr. Syst.*, vol. 27, pp. 5:1–5:20, 2022.

[10] Y. Du, Y. Gao, S. Huang, and Q. Li, “LDPC level prediction toward read performance of high-density flash memories,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 10, pp. 3264–3274, 2023.

[11] T.-Y. Wang, C.-W. Tsao, Y.-H. Chang, and T.-W. Kuo, “Retention-aware read acceleration strategy for LDPC-Based NAND flash memory,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 12, pp. 4597–4605, 2023.

[12] L. Cui, F. Wu, X. Liu, M. Zhang, and C. Xie, “VaLLR: Threshold voltage distribution aware LLR optimization to improve LDPC decoding performance for 3D TLC NAND flash,” in *2019 IEEE 37th International Conference on Computer Design (ICCD)*, 2019, pp. 668–671.

[13] W. Liu, F. Wu, M. Zhang, Y. Wang, Z. Lu, X. Lu, and C. Xie, “Characterizing the reliability and threshold voltage shifting of 3D Charge Trap NAND flash,” in *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2019, pp. 312–315.

[14] K.-K. Yong and L.-P. Chang, “Error diluting: Exploiting 3-D NAND flash process variation for efficient read on LDPC-Based SSDs,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 11, pp. 3467–3478, 2020.

[15] F. Wu, M. Zhang, Y. Du, W. Liu, Z. Lu, J. Wan, Z. Tan, and C. Xie, “Using error modes aware LDPC to improve decoding performance of 3-D TLC NAND flash,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 4, pp. 909–921, 2020.

[16] S. Ouyang, G. Han, Y. Fang, and W. Liu, “LLR-Distribution-Based non-uniform quantization for RBI-MSD algorithm in MLC flash memory,” *IEEE Communications Letters*, vol. 22, no. 1, pp. 45–48, 2018.