

# Exploiting Variable-Dimensional LDPC Coding to Improve NAND Flash Memory System Performance

Meng Zhang<sup>†</sup>, Wei Li<sup>†</sup>, Yangyi Li<sup>†</sup>, Tianwei Gui<sup>†</sup>, Changsheng Xie<sup>§</sup>, and Fei Wu<sup>§,\*</sup>

<sup>†</sup>School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

<sup>§</sup>Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074, China

\*Corresponding author: {Fei Wu, wufei@hust.edu.cn}

**Abstract**—Solid state drives (SSDs) based on NAND flash technology are steadily gaining popularity and mass market adoption due to their increased storage capacity and density. However, because of the more bits in each cell and the reduced cell spacing, they are experiencing a decline in reliability. The most efficient way to ensure reliability of data is to use low-density parity-check (LDPC) codes. Nevertheless, using a hybrid decoding technique for LDPC codes results in a significant decoding latency, which exacerbates performance issues. In this paper, we propose a variable-dimensional LDPC coding scheme, called VDLDP, to reduce the high decoding latency and thus improve read performance of NAND flash memory on hot read data. One of the crucial designs in the VDLDP scheme is the two-dimensional LDPC (TD-LDPC) algorithm. TD-LDPC implements row and column encoding separately when writing data to the flash memory by using sub-LDPC codes. Errors in the data arise after a period of retention. When the data is read out, TD-LDPC performs row and column decoding using sub-LDPC codes, and the column decoding result can be re-decoded as a new round of row decoding input. Simulation results show that the proposed VDLDP scheme has the advantage in decoding latency and reduces the flash memory read response time by up to 12.0% (5.8% on average across all workloads) compared to the current LDPC code scheme. The proposed VDLDP scheme ensures reliability while improving NAND flash system read performance on hot read data.

**Index Terms**—NAND flash memory, reliability, low-density parity-check (LDPC) codes, read latency.

## I. INTRODUCTION

Due to the advantages of small size, low energy consumption, and non-volatility, NAND flash-based solid state drives (SSDs) are gradually becoming the mainstream storage medium. To boost storage capacity and density, manufacturers are also consistently shrinking cell sizes and packing more data onto a single cell [1–3]. The increase in bits stored in a single cell from a 1 bit/cell single-level cell (SLC) to a 4-bit/cell quad-level cell (QLC) has caused the number of erroneous bits in flash memory to grow by more than ten times [4]. Crosstalk between cells hence poses a major threat to the reliability of data storage, even as flash memory capacity and storage density grow quickly.

Error correction code (ECC) is a useful technique for preventing interference from flash memory channels and guaranteeing the reliability of data stored [5, 6]. Commercial SSDs frequently employ traditional Bose–Chaudhuri–Hocquenghem (BCH) codes [7], yet they are unable to ensure the reliability of

data storage in high-capacity flash memory [8]. Furthermore, because of their superior error correction capacity, low-density parity-check (LDPC) codes [9] have drawn a lot of interest from both academia and industry [10–13]. High reliability is attained by LDPC codes by soft decoding, but at the expense of an increase in decoding latency, lowering NAND flash read performance [14, 15].

To improve NAND flash read performance with hot read data induced by decoding latency of LDPC codes, we propose a variable-dimensional LDPC (VDLDP) coding scheme in this paper. VDLDP consists of three main components: data pattern classifier, ECC scheme selector, and ECC module. One of the most important designs is the two-dimensional LDPC (TD-LDPC) algorithm in the ECC module. When the data is read out, TD-LDPC corrects the errors by performing row and column decoding of the sub-LDPC codes on the data. TD-LDPC fully utilizes the advantages of two-dimensional coding for fully parallel decoding. It starts with designing a single data word in two dimensions and corrects the errors of multiple rows and columns with sub-LDPC codes. To further expedite the decoding convergence, the row and column coordination scanning and decoding approach is also used throughout the decoding process. TD-LDPC is decoded using multiple rounds of iterative scanning, the output of the previous round of column decoding can be re-decoded as the input of the new round of row decoding until the end of decoding.

In summary, this paper makes the following contributions.

- We test the raw bit error rate (RBER) of real 3D triple-level cell (TLC) flash memory with different retention times and program/erase cycles (PEC). RBER variations between layers and pages motivate us to design high-performance LDPC codes.
- We propose the VDLDP scheme and explain its implementation scheme in detail. Innovatively, we propose to adopt the TD-LDPC algorithm to encode and decode the hot read data in the VDLDP scheme, and additionally give the iterative scanning characteristics during the decoding process.
- The effectiveness of VDLDP scheme and TD-LDPC algorithm is evaluated by numerical and system simulations. Results show that TD-LDPC can achieve full error correction at a raw bit error rate (RBER) of no more than

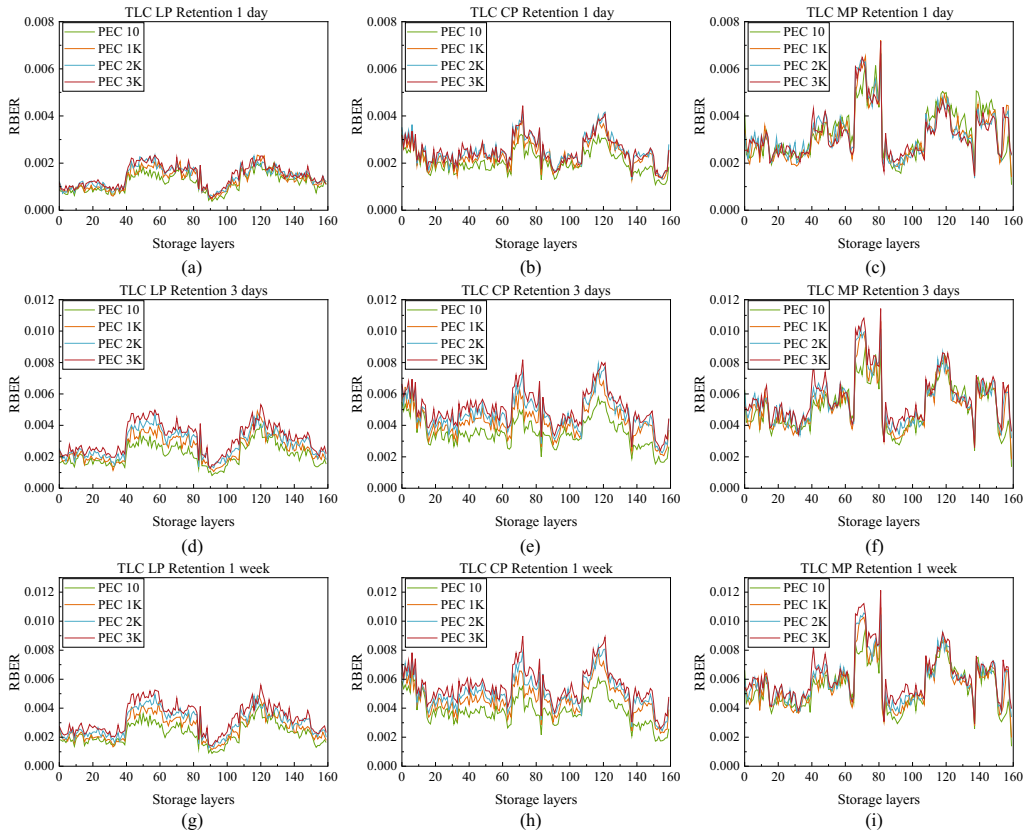


Fig. 1. The variations of RBER between 160 layers and three pages (i.e., LSB page(LP)(a,d,g), CSB page(CP)(b, e, h), MSB page(MP)(c, f, i)) under different PEC and retention time.

$5 \times 10^{-3}$ , and can achieve up to  $2\times$  and  $9\times$  reduction in decoding latency compared to hard LDPC and soft LDPC, respectively. Furthermore, compared to the current LDPC code scheme, VLDPC can reduce the read response time of NAND flash systems by up to 12.0% (5.8% on average across all workloads).

The rest of the paper is organized as follows. Section II gives the basic introduction of NAND flash memory with the host system, the basic principles of LDPC codes, and the related work. Motivation is introduced in Section III. The implement details of VLDPC and the encoding and decoding processes of TD-LDPC are presented in Section IV. Section V analyzes the results of numerical and system simulations. The paper is concluded in Section VI.

## II. MOTIVATION

We test the 3D TLC NAND flash memory chip with floating gate structure on a real FPGA hardware platform with the chip information shown in Table I. In our experiments, we use high temperature baking to accelerate the retention time. First, we generate a block size of random data. Second, we execute different numbers of PEC for different blocks, then uniformly bake the chip at high temperature. Finally, we count the difference between the output result and the input data. Through real experiments, we show the RBER of different

layers and pages of 3D TLC flash memory chips with different retention times (i.e., one day, three days, and one week) and PEC (i.e., 10, 1K, 2K and 3K) in Fig. 1.

The following findings can be obtained from Fig. 1. 1) RBER increases significantly with longer retention time or increased PEC. In comparison, retention time has a greater impact on RBER. 2) There are large differences in RBER between different layers, with relatively lower RBER between layers 0-40 and 80-100, and the higher RBER between layers 60-80. 3) There are also large differences in RBER between different pages, which basically manifests itself in the following way: LSB page < CSB page < MSB page. 4) In the case of a retention time of one day, the RBER is relatively low, basically remaining below  $5 \times 10^{-3}$ . When the retention time rises to three days or even one week, the RBER of the LSB page is still below  $5 \times 10^{-3}$  and the RBERs of the CSB page and MSB page stay below  $5 \times 10^{-3}$  for quite a number of layers under the condition that the PEC does not exceed 3K.

In conclusion, 3D TLC flash memory with low retention time maintains its RBER at a low level (e.g.  $5 \times 10^{-3}$ ) and increases slowly with increasing PEC. As a result, we characterize hot read data as information that the system reads often and has a comparatively short retention time in flash memory. There is significant potential for read speed enhancement because hot read data typically consumes a tiny amount of

TABLE I  
3D TLC NAND FLASH CHIP INFORMATION

Item	Specification
Logical units (LUNs) per chip	2
Planes per LUN	4
Blocks per plane	247
Pages per block	4800
Page size	18KB
Chip capacity	128GB
Layer number	160
Wordline number per layer	10

flash memory but is accessed frequently. Therefore, this paper proposes VLDLPC to enhance the system performance of hot read data.

### III. PROPOSED VLDLPC SCHEME

This section presents mainly details of the proposed VLDLPC scheme, which contains the encoding and decoding processes of TD-LDPC as well as the iterative scanning characteristic in the decoding process.

#### A. Overview

As the increase in read latency becomes a critical issue in high-density flash memory, we propose VLDLPC to improve read performance. The proposed VLDLPC system architecture is illustrated in Fig. 2. VLDLPC consists of three main components: a data pattern classifier, an ECC scheme selector, and an ECC module. From the reliability point of view, we pay more attention to the system read performance. Therefore, we categorize the read data patterns into hot and cold data. The data pattern classifier [14, 16] mainly distinguishes between hot and cold read data pages and passes them to the ECC scheme selector, which assigns ECC schemes to different pages. For hot read data pages, we introduce a two-dimensional LDPC (TD-LDPC) algorithm, which can significantly reduce the system read latency for hot read data and will be highlighted in the next subsection, otherwise, the traditional LDPC code scheme is normally applied.

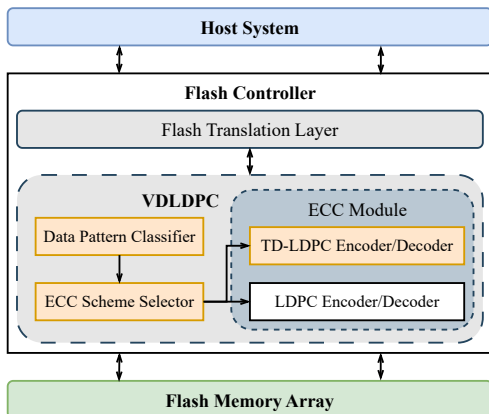


Fig. 2. The system architecture of the proposed VLDLPC.

#### B. TD-LDPC Design

TD-LDPC consists of two sub-LDPC codes, each of which has redundancy and information bits for each dimension. Suppose there are two subcodes:  $LDPC_1(n_1, j_1, k_1)$  and  $LDPC_2(n_2, j_2, k_2)$ , where  $n_i$ ,  $j_i$ , and  $k_i$  represent the number of columns, column weight, and row weight of the parity-check matrix of subcode  $i$  ( $i = 1, 2$ ), respectively. The column weight of a parity-check matrix refers to the number of non-zero elements in a column. Additionally,  $m_i$  is the number of rows of the parity-check matrix that satisfies  $m_i = n_i \times j_i / k_i$ ,  $j_i < k_i$ ,  $j_i \ll m_i$ ,  $k_i \ll n_i$ . Then the encoding steps of TD-LDPC are described as follows:

- Generate the information matrix: Construct an information matrix with a size of  $K$  information bits, divided into  $r$  rows and  $c$  columns. Make sure that the number of information bits in each row and column is distributed equally, so that  $n_1 - m_1 = K/r$  for rows and  $n_2 - m_2 = K/c$  for columns.
- Horizontal encoding: To produce the horizontal encoding parity-check matrix  $r \times m_1$ , encode the  $r$  rows of information bits using the subcode  $LDPC_1$  encoding method.
- Vertical encoding: To generate the vertical encoding parity-check matrix  $c \times m_2$ , encode the  $c$  columns of information bits using the subcode  $LDPC_2$  encoding method.

The information bit length is obtained by using formula (1).

$$K = r \times (n_1 - m_1) = c \times (n_2 - m_2) = \frac{1}{2} [r \times (n_1 - m_1) + c \times (n_2 - m_2)] \quad (1)$$

The code length is computed through formula (2).

$$N = K + r \times m_1 + c \times m_2 = \frac{1}{2} [r \times (n_1 + m_1) + c \times (n_2 + m_2)] \quad (2)$$

The code rate is calculated by exploiting formula (3).

$$R = K/N = \frac{r \times (n_1 - m_1) + c \times (n_2 - m_2)}{r \times (n_1 + m_1) + c \times (n_2 + m_2)} \quad (3)$$

Fig. 3 depicts the encoding structure of TD-LDPC, and every information bit is encoded both vertically and horizontally. As a result, every piece of information is double safeguarded, improving the data reliability.

The TD-LDPC code is constructed from independent horizontal and vertical LDPC subcodes; thus, decoding must also be executed separately for both. The decoding procedure consists of four steps: initialization of log-likelihood ratios (LLRs), check node update, variable node update, and hard decision, with the layered normalized min-sum (LNMS) algorithm applied. Given an encoded codeword  $c$  from message  $u$  and its received sequence  $r$ , correctness is verified by  $r \cdot H^T = 0$ . If this condition fails, LNMS iteratively updates the LLRs of check and variable nodes.

In practice, horizontal decoding is performed first, followed by vertical decoding. Since vertical decoding can recover additional bits, horizontal decoding is re-applied to

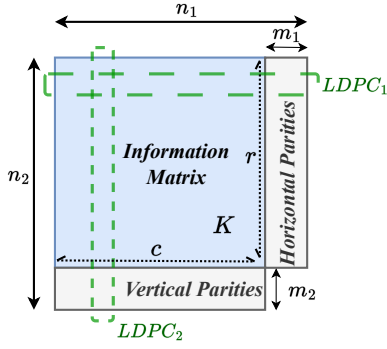


Fig. 3. The encoding structure of the proposed TD-LDPC scheme.

further enhance correction. This iterative scanning decoding scheme (Algorithm 1) begins by reshaping the flash-read bitstream into a matrix  $MsgRe$  containing information and parity symbols. The parameters include  $RowSize$  and  $ColSize$  (matrix dimensions),  $IterMax$  (maximum LNMS iterations), and  $CycleNum$  (maximum scanning cycles). Horizontal and vertical subcodes are decoded by  $Decoder1()$  and  $Decoder2()$  (lines 5 and 13), followed by matrix updates. If  $-1 < IterNum < IterMax$ , the row error is correctable; if  $IterNum = -1$ , the row contains no errors, and the counter of corrected rows is increased (line 8). The procedure terminates (lines 19–20) once no further correctable row or column errors remain, and the final decoded matrix  $MsgDe$  is output.

#### IV. EVALUATION RESULTS AND ANALYSIS

The simulation parameters used to assess the VDLDP are first given in this section. Simulation results are then presented and analyzed.

##### A. Simulation Setup

TABLE II  
EXPERIMENTAL PARAMETERS

Item	Specification
Flash capacity	192GB
Page size	4KB
Pages per block	512
Blocks per plane	2048
Sensing latency per page( $\mu s$ )	40 [17]
Transmission latency per byte( $ns$ )	5
Page program latency( $\mu s$ )	800
Block erase time( $ms$ )	3.5

We first perform numerical simulations to compare the error correction capability and decoding latency of TD-LDPC and current one-dimensional LDPC codes, and then utilize FEMU [18], an open-source multi-queue SSD emulator, to evaluate the impact of VDLDP and the current LDPC code scheme (introduced in Section II.B) on the system read performance under different flash chip states. The first procedure uses a quasi-cyclic parity-check matrix with a column weight of 4, a row weight of 36, and a cyclic permutation matrix (CPM) size of 1024 to encode and decode the 4KB data for LDPC

#### Algorithm 1 TD-LDPC Decoding

**Input:** The received message matrix  $MsgRe$ , the maximum number of cycles for iterative scanning  $CycleNum$ , row and column sizes of the initial information matrix  $RowSize$  and  $ColSize$ , the maximum number of iterations for the LNMS algorithm  $IterMax$

**Output:** The decoded message matrix  $MsgDe$

```

1: while  $CycleNum > 0$  do
2:   Initial  $cRowNum = 0$  and  $cColNum = 0$ ;
3:   /* Horizontal decoding */
4:   for  $i = 1$  to  $RowSize$  do
5:      $RowData, IterNum = Decoder1(MsgRe[i, :])$ ;
6:      $MsgRe[i, 1 : ColSize] = RowData$ ;
7:     if  $-1 < IterNum < IterMax$  then
8:        $cRowNum ++$ ;
9:     end if
10:  end for
11:  /* Vertical decoding */
12:  for  $j = 1$  to  $ColSize$  do
13:     $ColData, IterNum = Decoder2(MsgRe[:, j])$ ;
14:     $MsgRe[1 : RowSize, j] = ColData$ ;
15:    if  $-1 < IterNum < IterMax$  then
16:       $cColNum ++$ ;
17:    end if
18:  end for
19:  if  $cRowNum + cColNum == 0$  then
20:    break;
21:  end if
22:   $CycleNum --$ ;
23: end while
24:  $MsgDe = MsgRe[1 : RowSize, 1 : ColSize]$ ;

```

codes. We name the scheme with BF decoding algorithm as *LDPC-hard* and the scheme with LNMS decoding algorithm as *LDPC-soft*, respectively. The code rate is 8/9 and the maximum number of decoding 3iterations is set to 20 for both schemes. Table II primarily shows the parameter configurations used for the FEMU experiments. Our calculation of flash capacity takes into account a default overprovisioning ratio of 25%. We evaluate the impact of different schemes on the read performance of flash memory systems. The read-hot and read-cold workloads in real storage are separated by average overread [19]. The read ratio can be expressed as follows:  $total\ read\ size / user\ data\ size$ . We select eight read-hot workloads with a high read ratio for Table III from two benchmark suites that are frequently used to assess storage system performance: Microsoft Research Cambridge (MSRC) Traces [20] and Umass Trace Repository [21].

##### B. Results and Analysis of TD-LDPC

With a 4KB error correction unit, we design TD-LDPC schemes based on different sub-LDPC code lengths. The two-row, two-column scheme (*TD-LDPC-2ways*) employs identical row and column subcodes, each with column weight 4, row weight 68, and CPM size 256. The four-row, four-column

TABLE III  
STATISTICS OF WORKLOADS

Workload	Read	Write	Read ratio
src2_1	39749758464	193029120	99.5%
stg1	85387283456	6427211776	93.0%
web2	282199884288	841556480	99.7%
hm1	8878785024	580831744	93.9%
web1	4092185600	696815104	85.4%
rsrch2	558546944	310063104	64.3%
usr0	37903664640	14041345024	73.0%
WebSearch1	16368969728	0	100%

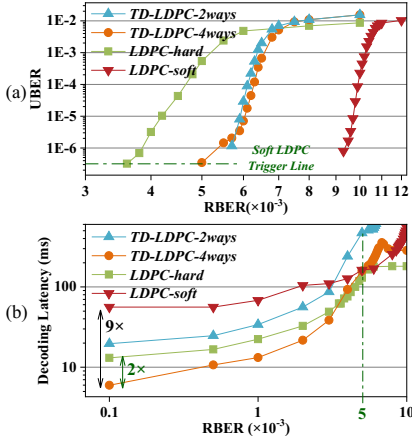


Fig. 4. The variations of UBER and decoding latency with different coding schemes under different RBERs. (a) RBER versus UBER. (b) RBER versus decoding latency.

scheme (*TD-LDPC-4ways*) adopts smaller CPM size 128. As baselines, we consider one-dimensional LDPC codes: *LDPC-hard* and *LDPC-soft*, which differ in decoding algorithms. Fig.4 compares the uncorrectable bit error rate (UBER) and decoding latency of these schemes under various RBERs. In Fig.4(a), *TD-LDPC-2ways* lowers RBER from  $6 \times 10^{-3}$  to  $1 \times 10^{-5}$ , while *TD-LDPC-4ways* reduces it from  $5 \times 10^{-3}$  to  $1 \times 10^{-6}$ . Both outperform *LDPC-hard* but remain weaker than *LDPC-soft*. The green dashed line denotes the soft LDPC trigger threshold of  $10^{-4}$  UBER, while the hard LDPC or TD-LDPC failure rate should not be greater than  $10^{-4}$  (i.e., 99.99%)[22]. TD-LDPC postpones the trigger to  $RBER = 5 \times 10^{-3}$ , compared with  $3.6 \times 10^{-3}$  for hard LDPC, thereby reducing latency. In Fig. 4(b), the green dashed line at  $RBER = 5 \times 10^{-3}$  marks the intersection of *TD-LDPC-4ways* and *LDPC-soft*. At this point, *TD-LDPC-4ways* reduces latency by up to 9× versus *LDPC-soft* and 2× versus *LDPC-hard*.

The error correction strength of TD-LDPC depends on that of its row and column subcodes. Although the subcodes have high rate (0.94) and relatively limited correction capability compared with *LDPC-soft*, TD-LDPC achieves stronger overall correction than the subcodes alone. Benefiting from soft decoding, TD-LDPC surpasses *LDPC-hard* while maintaining significantly lower latency than both baselines, due to parallel decoding of multiple rows or columns within each codeword.

### C. Results and Analysis of VLDLPC

From the previous experiment, we observe that *TD-LDPC-4ways* can reduce the decoding latency up to 9× and 2× compared to *LDPC-soft* and *LDPC-hard*, respectively. Then, when applying the TD-LDPC algorithm to the VLDLPC scheme, we are curious about the overall decoding latency performance on hot read data. Therefore, we select 160 layers and three pages of 3D TLC flash chips with the retention time of 1 day and PEC of 1K and 3K, respectively, to compare the difference in decoding latency between VLDLPC and the current LDPC code scheme (Baseline) implemented in flash memory, shown in Fig. 5. The difference between VLDLPC and Baseline is illustrated in Fig. 5(a) on the LSB page, where the decoding latency with the VLDLPC scheme is half of Baseline on average for both PEC of 1K and 3K. The decoding latency of most layers in Fig. 5(b) and (c) with the VLDLPC scheme is reduced by about 30% compared to the Baseline. The decoding latency of individual layers (more in Fig. 5(c)) with Baseline increases steeply because, as mentioned in the previous subsection, due to the existence of soft LDPC trigger lines, when the hard LDPC fails to resolve the error, it switches to the soft LDPC to continue the error correction, and the decoding latency of Baseline spikes at this point. In contrast, the VLDLPC scheme still does not cross the soft LDPC trigger line, so the decoding latency can be maintained at a relatively low level. In this case, the advantage of the VLDLPC scheme is even more obvious, with a maximum reduction of about 50% despite the high latency of the Baseline.

In addition, we are also curious about the performance improvement of VLDLPC on hot read data at the system level. Therefore, we select eight workloads to evaluate the improvement in system read response time for VLDLPC versus the current LDPC code scheme (Baseline). Furthermore, we evaluate the flash system response time model of the LSB page under various operating conditions: (a) PEC=1K, with 1 day of retention time; (b) PEC=2K, with 1 week of retention time; (c) PEC=3K, with 1 month of retention time. Additionally, we plot the flash system read response time displayed in Fig. 6.

We make three major observations from Fig. 6. First, the read performance of the simulated SSD is enhanced by the VLDLPC scheme. VLDLPC lower flash system read response time by up to 12.0% (5.8% on average across all workloads) compared to the baseline LDPC code. Second, the performance gain of VLDLPC decreases with worsening operating conditions within a certain range. For example, the average flash system read response time of VLDLPC across all workloads was 4.9% faster than the baseline LDPC code at 1K PEC and 1 day retention time. At 2K PEC and 1 week of retention time, this drops to 4.8%. However, the rate increased to 7.7% at 3K PEC and 1 month retention time. This is because for Baseline, the RBER has increased to the point where hard LDPC cannot handle it and has to be switched to soft LDPC, with which the system response time increases significantly. In contrast, VLDLPC can still handle errors at this point. Here is where VLDLPC comes in. Third, it does not mean that the lower

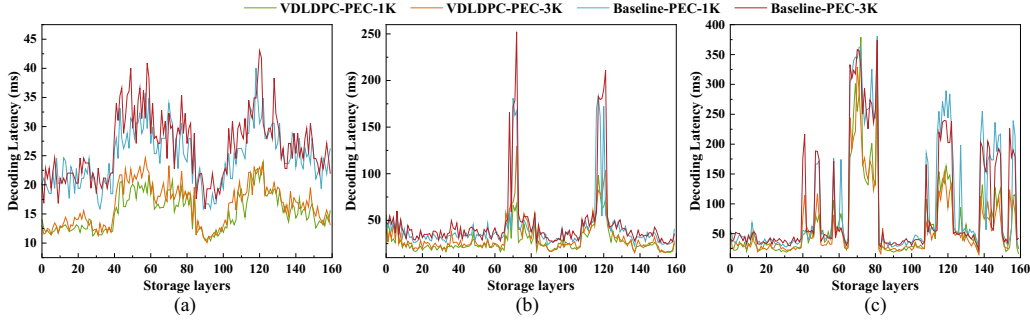


Fig. 5. The variation of decoding latency with VLDLPC and the Baseline between 160 storage layers and three pages (i.e., LSB page(a), CSB page(b), MSB page(c)) under the retention time of 1 day and PEC of 1K and 3K, respectively.

the read ratio, the worse the read performance of VLDLPC. VLDLPC improves the read performance of the flash system by 9.6% at 3K PEC and 1 month retention time, and by 5.1% on average under all operating conditions for rsrch2, whose read rate is only 64.3%.

Based on our observations, we conclude that VLDLPC can significantly reduce the decoding latency compared to the baseline LDPC code scheme while RBER is not higher than  $5 \times 10^{-3}$  and is effective at improving flash memory performance of hot read data by decreasing the decoding latency.

## V. CONCLUSION

This paper proposes a latency-efficient VLDLPC scheme, which effectively improves NAND flash memory performance of hot read data by reducing the decoding latency. We explain in detail the proposed VLDLPC scheme, the encoding and decoding procedures of TD-LDPC, and the iterative scanning characteristic in decoding. We evaluate the effectiveness of VLDLPC scheme and TD-LDPC algorithm using numerical and system simulations. Within the error correction capability of TD-LDPC, i.e., the RBER is not higher than  $5 \times 10^{-3}$ , the

decoding latency of TD-LDPC can be reduced by up to  $2\times$  and  $9\times$  compared to hard LDPC and soft LDPC, respectively. For the complete process of hot read data, the VLDLPC scheme can reduce the decoding latency by 30% or even 50% compared to the current LDPC code scheme. Using many real workloads, we can also observe that VLDLPC can lower flash system read response time by up to 12.0% (5.8% on average across all workloads). Therefore, the proposed VLDLPC scheme can effectively reduce decoding latency and improve flash system read performance of hot read data while ensuring reliability. VLDLPC can be well applied to SLC and TLC mixed mode SSD, especially SLC as a cache application scenario, when the main focus is on accessing hot read data, with high requirements for read performance.

## ACKNOWLEDGEMENTS

This work was supported in part by the Joint Fund for Intelligent Computing of Shandong Natural Science Foundation under Grant ZR2023LZH007 and ZR2024LZH004; in part by the National Natural Science Foundation of China under Grant U22A2071 and Grant 62372197; and in part by the Key Laboratory of Information Storage System, and Engineering Research Center of Data Storage Systems and Technology, Ministry of Education of China.

## REFERENCES

- [1] J. Cui, F. Chen, L. Li, S. Nie, and L. T. Yang, "Smart-NetSSD: exploiting path resources for read performance improvement in network-based SSDs," in *ICCD*, 2024, pp. 356–359.
- [2] Q. Xia and W. Xiao, "Improving MLC flash performance with workload-aware differentiated ECC," in *ICPADS*, 2016, pp. 545–552.
- [3] S.-Q. Nie, C. Zhang, and W.-G. Wu, "DIR: dynamic request interleaving for improving the read performance of aged solid-state drives," *JCST*, vol. 39, no. 1, pp. 82–98, 2024.
- [4] X. Yu, J. He, B. Zhang, X. Wang, Q. Li, Q. Wang, Z. Huo, and T. Ye, "Interleaved LDPC decoding scheme improves 3D TLC NAND flash memory system perfor-

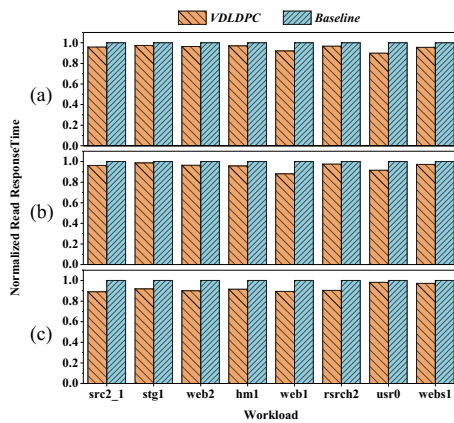


Fig. 6. Read response time comparison of the VLDLPC scheme and Baseline under different PEC and retention time, (a) PEC=1K, with 1 day of retention time; (b) PEC=2K, with 1 week of retention time; (c) PEC=3K, with 1 month of retention time.

- formance,” *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.*, no. 11, pp. 4191–4204, 2023.
- [5] J. Yang, “ECC approach for correcting errors not handled by RAID recovery,” in *FMS*, Santa Clara, CA, USA, 2017.
- [6] J. Cui, C. Liu, J. Liu, J. Huang, and L. T. Yang, “Exploiting uncorrectable data reuse for performance improvement of flash memory,” *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol. 41, no. 6, pp. 1785–1798, 2021.
- [7] R. C. Bose and D. K. Ray-Chaudhuri, “On a class of error correcting binary group codes,” *Information and control*, vol. 3, no. 1, pp. 68–79, 1960.
- [8] K. Zhao, W. Zhao, H. Sun, X. Zhang, N. Zheng, and T. Zhang, “LDPC-in-SSD: making advanced error correction codes work effectively in solid state drives,” in *FAST*, 2013, pp. 243–256.
- [9] R. Gallager, “Low-density parity-check codes,” *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, 1962.
- [10] H. Feng, D. Wei, S. Gu, Z. Piao, Y. Wang, and L. Qiao, “Random flip bit aware reading for improving high-density 3-D NAND flash performance,” *TCAS-I*, 2024.
- [11] 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 *ICCD*, 2019, pp. 668–671.
- [12] B. Bao, Q. Li, W. Guan, Q. Wang, L. Liang, and X. Qiu, “Adaptive granularity progressive LDPC decoding for NAND flash memory,” *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol. 43, no. 4, pp. 1312–1316, 2023.
- [13] 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 Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol. 42, no. 12, pp. 4597–4605, 2023.
- [14] Y. Du, Q. Li, L. Shi, D. Zou, H. Jin, and C. J. Xue, “Reducing LDPC soft sensing latency by lightweight data refresh for flash read performance improvement,” in *DAC*, 2017, pp. 1–6.
- [15] 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 Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol. 41, no. 3, pp. 544–557, 2021.
- [16] W.-L. Wu, J.-W. Hsieh, and H.-Y. Ku, “CDS: coupled data storage to enhance read performance of 3D TLC NAND flash memory,” *IEEE Trans. Comput.*, vol. 73, no. 3, pp. 694–707, 2023.
- [17] S. Cho, B. Kim, H. Cho, G. Seo, O. Mutlu, M. Kim, and J. Park, “AERO: adaptive erase operation for improving lifetime and performance of modern NAND flash-based SSDs,” in *ASPLOS*, vol. 3, 2024, pp. 101–118.
- [18] H. Li, M. Hao, M. H. Tong, S. Sundararaman, M. Bjørling, and H. S. Gunawi, “The CASE of FEMU: cheap, accurate, scalable and extensible flash emulator,” in *FAST*, 2018, pp. 83–90.
- [19] M. Fukuchi, S. Suzuki, K. Maeda, C. Matsui, and K. Takeuchi, “BER evaluation system considering device characteristics of TLC and QLC NAND flash memories in hybrid SSDs with real storage workloads,” in *ISCAS*, 2021, pp. 1–4.
- [20] MSR, “MSR cambridge traces,” 2009, <http://iotta.snia.org/traces/388>.
- [21] UMass, “UMass trace repository,” 2007, <http://traces.cs.umass.edu/index.php/Storage/Storage>.
- [22] Y. Cai, S. Ghose, E. F. Haratsch, Y. Luo, and O. Mutlu, “Error characterization, mitigation, and recovery in flash-memory-based solid-state drives,” *Proc. IEEE*, vol. 105, no. 9, pp. 1666–1704, 2017.
- [23] R. Nadig, M. Sadrosadati, H. Mao, N. M. Ghiasi, A. Tavakkol, J. Park, H. Sarbazi-Azad, J. G. Luna, and O. Mutlu, “Venice: improving solid-state drive parallelism at low cost via conflict-free accesses,” in *ISCA*, 2023, pp. 1–16.
- [24] Y. Li, G. Han, C. Liu, M. Zhang, and F. Wu, “Exploiting the single-symbol LLR variation to accelerate LDPC decoding for 3-D NAND flash memory,” *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol. 42, no. 12, pp. 5146–5150, 2023.
- [25] 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 Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol. 39, no. 4, pp. 909–921, 2019.
- [26] Y. Wang, D. Wei, M. Liu, H. Feng, and L. Qiao, “EBDN: entropy-based double nonuniform sensing algorithm for LDPC decoding in TLC NAND flash memory,” *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol. 43, no. 6, pp. 1914–1918, 2024.
- [27] Y. Song, Y. Lv, and L. Shi, “DECC: differential ECC for read performance optimization on high-density NAND flash memory,” in *ASP-DAC*, 2023, pp. 104–109.
- [28] 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,” *TCAS-I*, 2024.
- [29] Y. Li, G. Han, S. Huang, C. Liu, M. Zhang, and F. Wu, “Exploiting metadata to estimate read reference voltage for 3-D NAND flash memory,” *IEEE Trans. Consum. Electron.*, vol. 69, no. 1, pp. 9–17, 2022.
- [30] J. Cui, Z. Zeng, J. Huang, W. Yuan, and L. T. Yang, “Improving 3-D NAND SSD read performance by parallelizing read-retry,” *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol. 42, no. 3, pp. 768–780, 2022.
- [31] X. Fang, M. Zhang, Y. Guo, F. Chen, B. Chen, X. Zhan, J. Wu, F. Wu, and J. Chen, “High-precision short-term lifetime prediction in TLC 3-D NAND flash memory as hot-data storage,” *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol. 42, no. 10, pp. 3224–3235, 2023.