



Article

A Low-Cost Improved Method of Raw Bit Error Rate Estimation for NAND Flash Memory of High Storage Density

Kainan Ma ^{1,2} , Ming Liu ^{1,2,*}, Tao Li ^{1,2}, Yibo Yin ^{1,2}  and Hongda Chen ^{1,2}¹ Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100864, China;

makainan@semi.ac.cn (K.M.); litao@semi.ac.cn (T.L.); yyb2018@semi.ac.cn (Y.Y.); hdchen@semi.ac.cn (H.C.)

² University of Chinese Academy of Sciences, Beijing 100049, China

* Correspondence: liuming@semi.ac.cn

Received: 2 October 2020; Accepted: 9 November 2020; Published: 12 November 2020



Abstract: Cells wear fast in NAND flash memory of high storage density (HSD), so it is very necessary to have a long-term frequent in-time monitoring on its raw bit error rate (RBER) changes through a fast RBER estimation method. As the flash of HSD already has relatively lower reading speed, the method should not further degrade its read performance. This paper proposes an improved estimation method utilizing known data comparison, includes interleaving to balance the uneven error distribution in the flash of HSD, a fast RBER estimation module to make the estimated RBER highly linearly correlated with the actual RBER, and enhancement strategies to accelerate the decoding convergence of low-density parity-check (LDPC) codes and thereby make up the rate penalty caused by the known data. Experimental results show that when RBER is close to the upper bound of LDPC code, the reading efficiency can be increased by 35.8% compared to the case of no rate penalty. The proposed method only occupies 0.039 mm² at 40 nm process condition. Hence, the fast, read-performance-improving, and low-cost method is of great application potential on RBER monitoring in the flash of HSD.

Keywords: low-density parity-check (LDPC) code; NAND flash memory; parameter estimation

1. Introduction

NAND flash memory technology has been flourishing since the first flash memory being invented by Dr Fujio Masuoka [1] in 1984. The development of NAND technology has brought tremendous changes to the memory market and the electronics industry. However, the cost of flash memory still needs to be reduced to gain wider acceptance in mass storage by increasing the storage density. The innovation of etching technology has made NAND flash develop from 2D structure [2] to 3D structure [3], and the application of incremental step pulse program (ISPP) scheme [4,5] has realized the precise program voltage control, which makes it possible to increase the storage density of NAND flash. The density of storage has been rising from 2D-256Kb single-level cell (SLC) [6], 2D multi-level cell (MLC) [7], and 3D MLC [8], to the current 3D-768Gb triple-level cell (TLC) [3], 3D quad-level cell (QLC) [9,10], and the under-developed penta-level cell (PLC). NAND cells can be designed as n bits/cell by precisely controlling the levels of the program threshold voltage V_{th} , where SLC to PLC stores 1–5 bits/cell corresponding to 2^1 – 2^5 voltage levels respectively. The high-storage-density NAND flash memories like QLC and PLC start to receive much attention due to the high demands on storage capacity and the low cost. However, the increase in storage density makes the data stored in these NAND flash memories vulnerable to noise interference [11–14].

Some error control strategies such as decoding statuses selection [15], ANN-coupled decoding [16], retention optimized reading [13] are of high potential to be applied in the NAND flash memory of

high storage density (HSD) to improve its error performance and extend its lifetime, considering that it has more errors and shorter lifetime as the storage density gets higher. The idea is to adopt different strategies for low raw bit error rate (RBER) and high RBER respectively, including using different decoding algorithms for low and high RBER, inputting more accurate soft information to the decoder based on RBER change learned with artificial neural networks (ANN), shifting read reference voltages V_{rs} or configuring multi-level V_{rs} in read operation when RBER goes high due to V_{th} shift. Therefore, having RBER estimated would be a great help to these strategies for memory management and wearing level monitoring. As RBER grows fast in HSD NAND flash, it is very necessary to frequently monitor the RBER in a long term so that the error control methods can be executed at the right time to maintain the performance of the flash. Ref. [17] proposed a method of using parity violation to estimate RBER, and it needs to multiply a pre-set quantized scalar with the number of parity violations, which is a function of the number of parity violations and determines the accuracy of the estimated BER. However, considering that parity violation can only reflect the odd number of errors in a codeword, it needs a large sample size N to ensure accurate estimation. For example, it needs to collect $N = 2880$ for a 16 KB-page-size flash, which is a very time-consuming collection. The pre-set parameter may not be able to cope with the complex and changing noise condition in HSD flash in time. Hence, it is of hysteresis for HSD NAND flash, where the noise, including Program/Erase (P/E) cycle effects [12], retention error [13], and read disturb error [14], will cause RBER increasing quickly with reading times or P/E cycles and the quickly changed error condition that happens during a read operation is necessary to be measured and reflected in time. Ref. [18] proposed a fast estimation method by comparing the test data with the data read from the flash memory, but it may not be so suitable for HSD flash memory because frames from different pages have unbalanced error distribution due to data modulation, which can cause inaccuracy in estimation. Meanwhile, the extra redundancy caused by test data reduces the efficiency of reading, especially for the HSD flash which has relatively lower speed but needs the long-term, frequent and in-time RBER monitoring. The read speed of the flash has already been lower than MLC and TLC due to the higher storage density, so to avoid further slowing down, the RBER estimation method should not slow down or even would be better to speed up the reading process.

In this paper, we proposed an improved and easy-to-implement fast RBER estimation method for HSD NAND flash utilizing true-value data comparison, which can also strengthen error performance of error control code (ECC), thereby increasing reading efficiency of the flash. The method includes an interleaving module to balance the errors from different pages, an RBER estimation module, and enhancement strategies for hard-decision and soft-decision decoding to improve error performance.

The rest of the paper is organized as follows. Section 2 explains how the data modulation affects the error distribution on pages and how the interleaving module alleviates the unbalance. Section 3 describes the procedure and parameter setting of the RBER estimation. Section 4 depicts the strategies of improving error performance of ECC by using extrinsic information from the redundant true-value data. Section 5 presents a hardware implementation of the proposed method. Section 6 draws a conclusion.

2. Data Modulation in NAND Flash

Data modulation refers to how to map the bit codes to a voltage level programmed into a NAND flash cell. The distribution of bit errors in the NAND flash highly depends on the pattern of data modulation. The currently widely used data mapping method is 1/2 division gray coding, or 1/2 gray coding for short. The term “1/2 division” means that the first line of the mapping table is divided by “1” and “0” in half, and each line half-divides the previous line by “1” and “0” again, as shown in Figure 1a. Another pattern is balanced gray coding, which is mentioned in [19] as an alternative for QLC data modulation. The term “balanced” means that each line of the mapping table has almost the same times of toggling between “1” and “0”. Since the errors in the flash is mainly due to the V_{th} in the

cell being shifted to the neighbouring levels by the noise interference, using gray coding can minimize the number of bit errors caused by the shift.

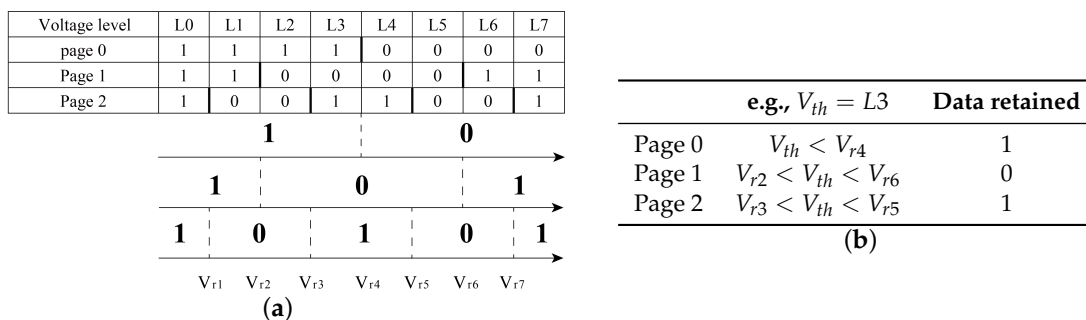


Figure 1. (a) An example of read reference voltages in triple-level cell (TLC) with 1/2 gray coding modulation; (b) how the data is demodulated from a TLC NAND flash cell.

Data are programmed or erased in the NAND flash cells through the Fowler–Nordheim (FN) tunnelling effect [20]. The level of the program voltage can be controlled by how much charge the cell stores since the program voltage is proportional to the tunnelling charge density in the cell. When the 1/2 gray coding data modulation is used, the data are stored via ISPP, which increases the threshold voltage V_{th} step by step to charge a cell to a certain program voltage level. The tunnelling electrons of the cell are gradually stored and accumulated to a certain level as required. When the balanced gray coding method is used, the data are stored in a cache first, and the cells are then charged to the voltage levels by incremental steps or in one pass.

Data are read from cells by comparing V_{th} with V_r s on each page to demodulate them to corresponding codes, as shown in Figure 1a,b. However, the widths of discriminant intervals are different on pages and wider interval has a higher tolerance to errors caused by the shift of V_{th} . Therefore, the frames from the upper page have higher frame error rates (FER) than the lower page. The experimental data of MLC in [21] shows that the error rate on the upper page has already been slightly higher than the lower page. Although the difference is not much problematic in MLC because of its low storage density, it will be in QLC and PLC as the storage density goes higher and FER of the most upper page can become huge, as shown in Figures 2 and 3. The unbalanced page error rates (PER) will cause inaccuracy in RBER estimation. The balanced gray coding modulation has less unbalanced PER, but there is still at least one page having higher PER than the others. We use PLC to illustrate the situation because it is the flash of the highest storage density currently known, so the situation is more obvious on it. As shown in Figure 2, the PERs are hugely unbalanced when 1/2 gray coding modulation is used, and frames from page 4 have FERs about 2.5 times as high as the RBER of the NAND flash. Meanwhile, the errors grow much faster on upper pages that the slope of page 4 is 9.7 times of page 0. Consequently, a very large FER will appear on page 4, causing inaccurate estimation as the sample frames are from different pages. When the balanced modulation is used, in Figure 3, the PERs are still not completely balanced that the frames from page 4 still have FERs about 1.5 times of the RBER, which will also affect the accuracy.

The accuracy of RBER estimation will be affected by whether every frame sampled for the RBER estimator having similar FER. Since both types of modulation are possibly used in HSD flash, it is important that the RBER estimation method can be compatible with them both to balance the errors.

To meet this requirement, an interleaving module is applied to alleviate the effect of the data modulations and making FERs of each frame as equally as possible, in order to achieve more accurate estimation. Interleaving is to swap the places of the message bits in the frames before modulating them to voltage levels. Correspondingly, a deinterleaving module is applied to restore the swapped message bits after the frames are retained. In our implementation, we built a uniformly randomly generated lookup table to allocate each bit to a specific position and the same lookup table will be used for all frames. For example, message bit 1 is allocated to position 5, bit 2 is allocated to position 11, and so on.

The interleaving aims to spread the message bits evenly across the memory cells. The deinterleaving restores the original order of the message bits by using the lookup table reversely. For instance, the bit in position 5 is back to message bit 1, and the bit in position 11 is back to bit 2, and so on. The interleaving scheme is not unique, one can apply other interleaving schemes for his own design purposes. The effect of interleaving depends on the length of the interleaved message bits and can be measured with the variance of FERs. As shown in Figure 4, the effect is better when the interleaved message bits become longer, but the variance drops slowly when the length exceeds 500 and almost no longer drops when the length exceeds 1200. Considering that the longer interleaving requires larger read-only memory and takes more clock cycles, which will increase the hardware complexity and extend processing time, it is recommended that interleaving a message in every 1024 bits is sufficient. As shown in Figures 5 and 6, for both types of data modulation, interleaving makes the disturbed bits evenly distributed to each frame and turns the increasing rates of FERs from each page to almost the same.

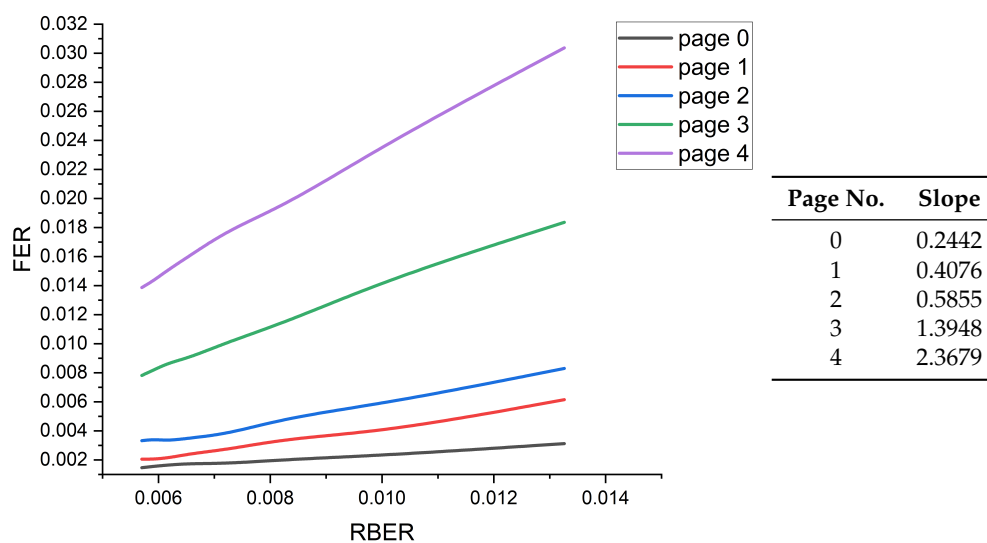


Figure 2. Frame error rates (FERs) of frames from different pages using 1/2 gray code modulation.

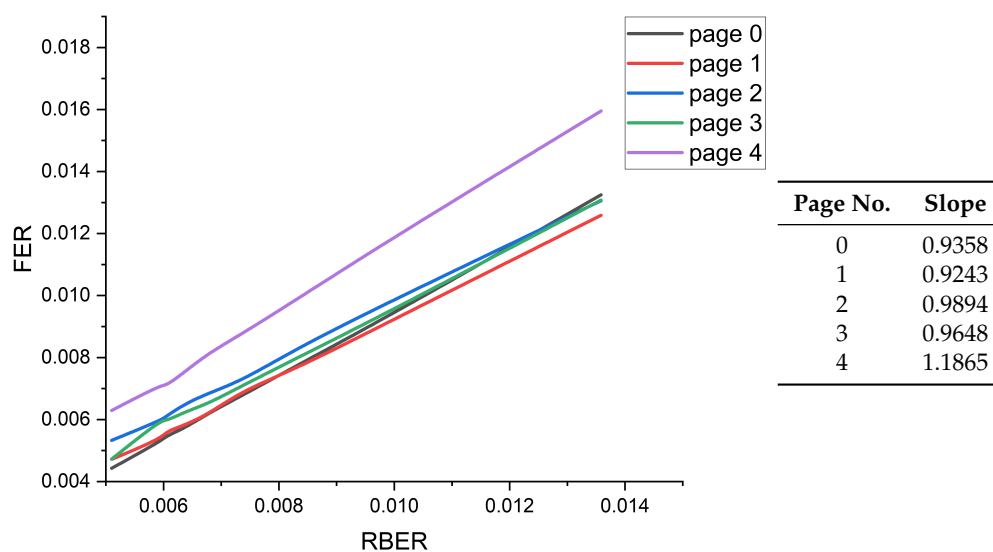


Figure 3. FERs of frames from different pages using the balanced gray coding modulation.

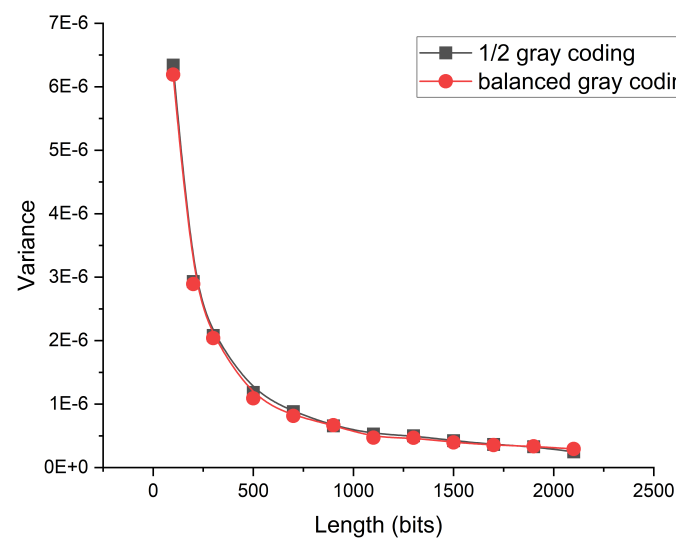


Figure 4. The variance of page error rates vs. lengths of interleaved messages bits tested at RBER = 0.01.

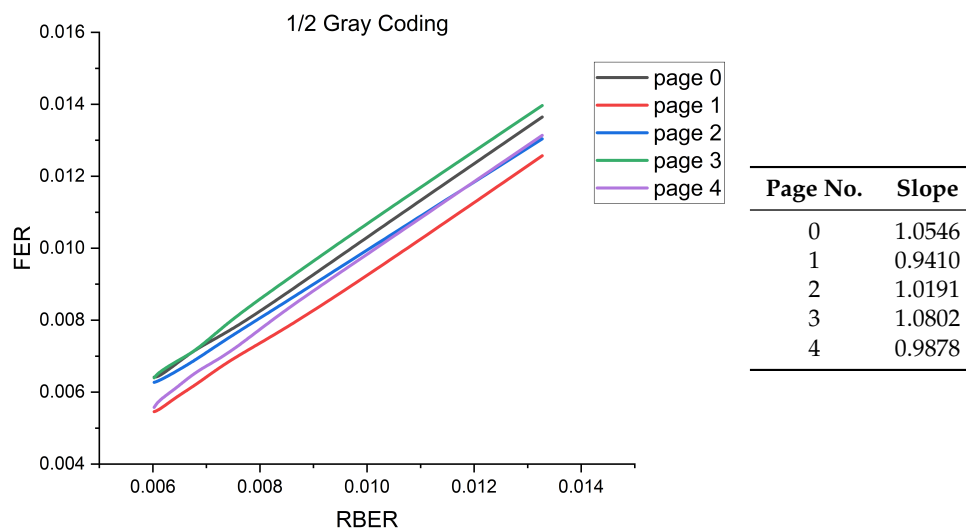


Figure 5. FERs of frames from different pages using 1/2 gray coding modulation with interleaving.

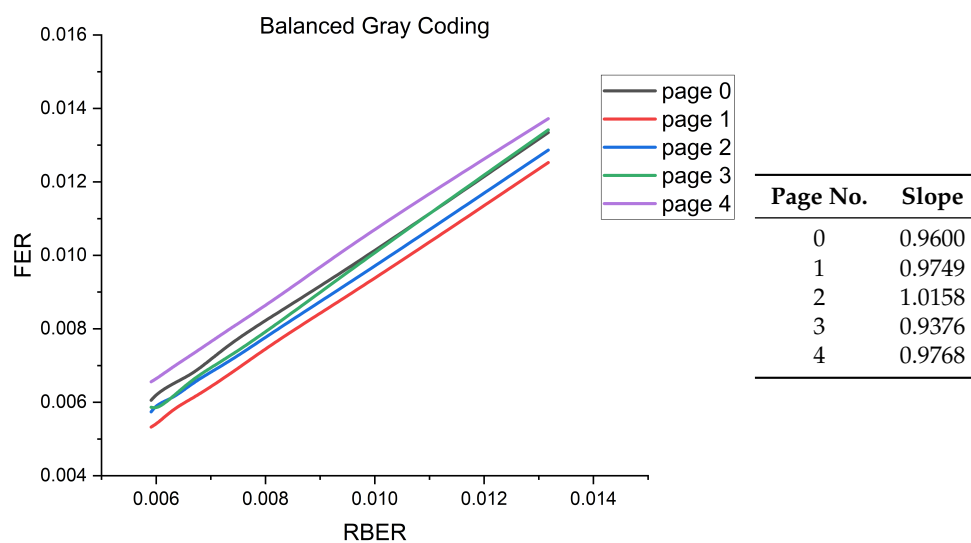


Figure 6. FERs of frames from different pages using balanced gray coding modulation with interleaving.

3. RBER Estimation

The RBER is estimated by comparing the true-value sequence inserted into specific positions of each frame with the one read from the flash to get an estimated FER. To ensure the accuracy of the estimated RBER, averaging the estimated FERs of multiple sample frames is required. The accuracy depends on two factors, the length of the true-value sequence L and the number of sample frames N . With the interleaving module to balance the errors of frames from different pages, there will be no huge difference in FERs obtained from the frames. Otherwise, it may result in inaccurate estimation in HSD NAND flash.

We estimated RBER with the mean of $N = 16$ sample frames because the division by 16 in estimation can be implemented with a right shifter. Moreover, it is a short time to wait and collect 16 frames in the HSD NAND flash, especially when the parallel structure of encoding and decoding is used, so the RBER of the flash can be reflected in time. The larger N can certainly lead to more accurate estimation, but it will take longer. The length L of the true-value sequence should also be selected carefully because a too-long sequence will cause too much rate penalty, whereas a too-short one will cause inaccurate estimation. Figure 7 shows how the lengths of the true-value sequence affect the mean square error (MSE) between \overline{BER}_{est} , the estimated BER, and the actual RBER when tested with a (10,080, 8400) QC-LDPC from [22]. The MSE is about 1.7×10^{-5} at the length of 200/8400, about 5 times larger than 3.4×10^{-6} at 300/8400, and then stays at the same order of magnitude down to 2.9×10^{-6} at 400/8400. Namely, the MSE is decreasing with the increase of the length, but the trend is gradually slowing down when the length exceeds 300/8400. To balance the rate penalty and the accuracy of estimation, the length is suggested to be less than 3.5% of the total length of the frame. We set $L = 256$ in our test for the convenience of hardware implementation. Figure 8 illustrates two cases of estimation at RBER = 0.01 and 0.1. In the HSD flash using 1/2 gray coding modulation, the estimation is very inaccurate when directly comparing data without interleaving. In this case, only 55.49% and 36.99% of the estimated values fall in [0.009, 0.011] and [0.09, 0.11], respectively, the $\pm 10\%$ range of the actual RBER. The interleaving module greatly improves the accuracy of the estimation for such flashes, making 85.37% and 99.55% of the estimated values fall within [0.009, 0.011] and [0.09, 0.11], respectively, and significantly narrowing the distribution interval. For the flash using balanced gray coding modulation, where about 83.4% and 97.89% of the estimated values are located within [0.009, 0.011] and [0.09, 0.11], respectively, interleaving only slightly increases the accuracy and narrows the distribution interval. Nevertheless, in general, interleaving can improve the accuracy of RBER estimation in HSD NAND flash using either modulation. Figure 9 shows that with the interleaving, the method can make \overline{BER}_{est} very close to the actual RBER, and they are highly linearly correlated though there exist some small fluctuations. Therefore, it is credible to estimate RBER in NAND flash with the true-value insertion method equipped with interleaving, and the good linearity can make the \overline{BER}_{est} sensitive to RBER change. Hence, it is suitable to measure and reflect RBER in HSD NAND flash in time.

\overline{BER}_{est} can help select decoding statuses. Usually, the soft-decision decoding is selected after the default reading level with hard-decision decoding and retry level with hard-decision decoding are failed, which wastes time on trial and error. Based on the current BER estimated, the decoding status can be directly selected so that the controller does not have to try through all decoding statuses, which thereby increase the reading speed for NAND flash. To realize this, a threshold RBER BER_{th} can be set to trigger switching between hard-decision and soft-decision decoding statuses that hard-decision decoding is applied when $\overline{BER}_{est} < BER_{th}$, and the soft-decision decoding is used otherwise. The method can also be used for cell wearing monitoring. The RBER will be rising when NAND flash cells are gradually wearing out as the number of P/E cycles increasing, so the long-term recorded \overline{BER}_{est} s can reflect the cell wearing condition in NAND flash. A block is possibly being worn out when its \overline{BER}_{est} reaches a certain threshold, and the controller should be informed to execute error management strategies or migrate the data to other blocks.

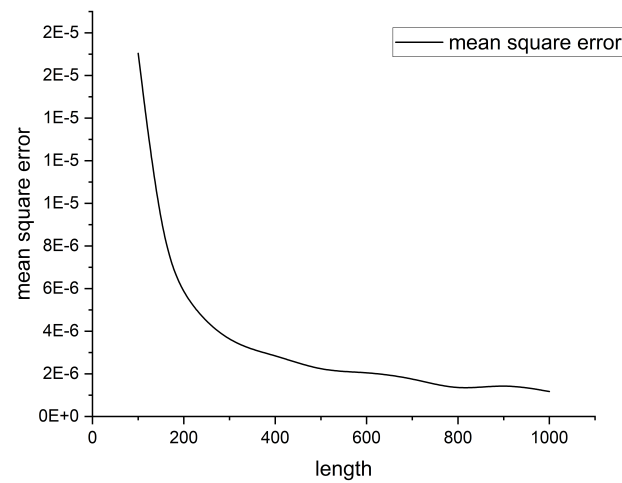


Figure 7. Mean square errors between the actual raw bit error rate (RBER) in NAND flash and the RBER estimated with the true-value sequence of different lengths, tested in RBER = (0, 0.1] with $N = 16$.

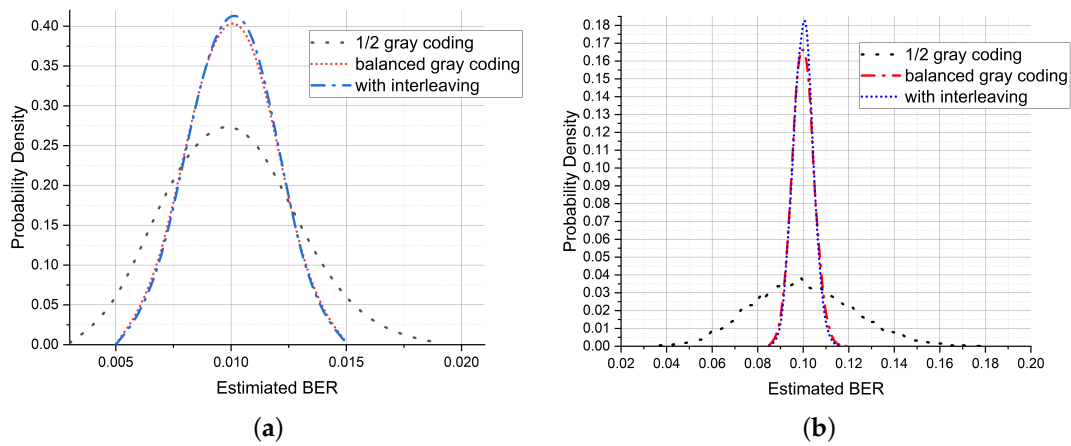


Figure 8. Probability density of estimated BER at actual (a) RBER = 0.01 and (b) RBER = 0.1 when using 1/2 gray coding modulation, balanced gray coding modulation, and with interleaving.

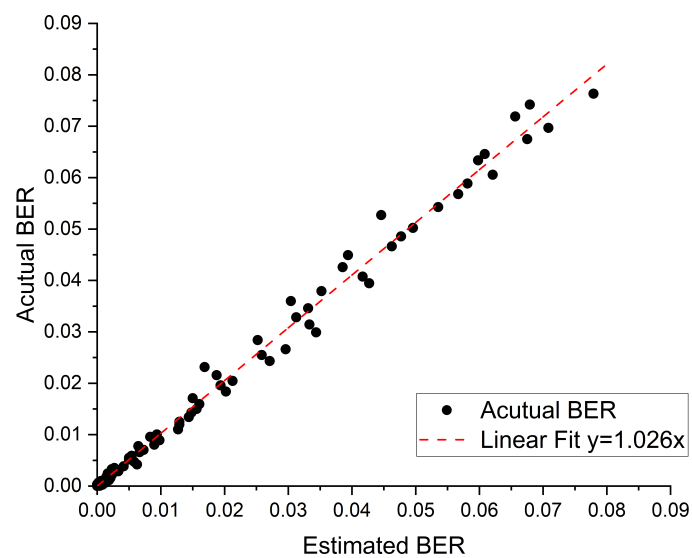


Figure 9. The linear fit of the RBER estimated with $L = 256$ vs. the actual RBER, where the coefficient of determination $R^2 = 0.992$.

4. Enhancement Strategy for Decoding

Since the HSD NAND flash has high storage space but relatively lower read speed, we care more about reading performance than the extra storage space taken up. We hereby proposed the enhancement strategies to accelerate decoding convergence with extrinsic information from the redundancy, making up the rate penalty caused by it.

To detect and correct errors in HSD NAND flash, the low-density parity-check codes (LDPC) [23,24] are usually used, which are suitable for scenarios that the code rates are higher than 2/3 [25]. Typically, the code rate in NAND flash is above 0.8 and even reaches 0.95, and the block length is also very large, up to 8 Kbit or even 20 Kbit [26–28]. Moreover, the encoding and decoding of LDPC can be easily implemented in parallel to increase throughput [22,29–31]. Among types of LDPC codes, quasi-cyclic LDPC (QC-LDPC) code [32] is a structured LDPC code recommended because it satisfies the row/column constraint to make sure no loop iteration in decoding which otherwise can result in decoding failure [33–35]. The decoding of QC-LDPC can have no error floor down to 10^{-10} [29].

There are two kinds of LDPC decoding algorithms, hard-decision decoding and soft-decision decoding. The hard-decision directly converts received symbols into demodulated bits “0” or “1”, whereas the soft decision is based on the probability of received symbols, expressed using logarithmic likelihood rate (LLR), to decide the most likely value for the corresponding bit. LLR is presented as

$$L_{v_j} = \log \frac{\Pr(\hat{y}_j = 0 | y_j)}{\Pr(\hat{y}_j = 1 | y_j)}, \quad (1)$$

where L_{v_j} is the LLR value, \hat{y}_j is the transmitted bit, and y_j is the received bit. The posterior probability $\Pr(\hat{y}_j | y_j)$ is determined by BER of the received codeword. The hard-decision decoding algorithms include majority logic decoding and bit flip decoding [23], which are characterized by low complexity, fast speed, but weak error correction capability. The most widely used soft-decision decoding algorithms are sum-product algorithm (SPA) and minimum sum algorithm (MSA), a simplification of SPA [24,30]. The computational complexity of soft-decision algorithms is much higher than the hard-decision ones, but their error-correction capability is stronger.

Figure 10 represents our enhancement strategies for bit-flip decoding and SPA decoding. The content and position of the true-value sequence are known. For bit flip decoding, as in Figure 10a, denote the position of true-value sequence as loc_{tv} , the number of parity violation for each bit of the read codeword as $f_{i,i=1,2,3,\dots}$, the positions of the maximum in f as loc_{pv} , and the position of bit flipping as loc_{bf} . Before decoding starts, fill back the true values to the read codeword. Then, calculate the number of parity violation for each bit. Find the positions where the maximum number of parity violations are, and remove the positions of the true values if there is any to get the position of the bits to be flipped loc_{bf} . While loc_{bf} is null, give loc_{bf} the positions of the second maximum in f . Repeat the above procedures until there is no parity violation or the program reaches its max iterations. As shown in Figure 11, the enhancement strategy gives the bit flip decoding better error performance, so more errors can be corrected by hard decision in the same number of iterations. Therefore, BER_{th} can be slightly increased in switching decoding statuses accordingly, and more decoding can use the hard-decision algorithm. The flash reading speed is thus increased at lower RBER. For SPA decoding, as Figure 10b, denote $v_{j,j=1,2,3,\dots}$ as variable node j , $c_{i,i=1,2,3,\dots}$ as check node i , L_{v_j} as the LLR of variable node j , $L_{v_j-c_i}$ as the value passed by variable node j to check node i , and $m_{c_i-v_j}$ as the value passed by check node i to variable node j . Before decoding starts, the LLRs at the positions of the true values are set to very large values, usually at least 10 times larger than the others, because we have a strong belief that the true values backfilled are correct. The LLRs will be propagated as the initial values of L_{v_j} and $L_{v_j-c_i}$. $m_{c_i-v_j}$ is updated with $L_{v_j-c_i}$ of each v_j connected to c_i , and then $L_{v_j-c_i}$ is updated with $m_{c_i-v_j}$ of each c_i connected to v_j . The L_{v_j} is then updated with $L_{v_j} = \sum_{i=1}^M L_{v_j-c_i}$, where M is the total number of check nodes. Then \hat{y}_j is discriminated based on L_{v_j} and parity check is done for \hat{y} . The above procedures are repeated until there is no parity violation, or the program reaches its max

iterations. As shown in Figure 12, with the enhancement strategy, the SPA decoding achieves better error performance and converges faster at higher RBER, which means that some iterations can be saved in decoding. The iteration of soft-decision decoding is time-consuming, compared with hard-decision decoding, so saving iterations can certainly improve the read speed, especially at higher RBER.

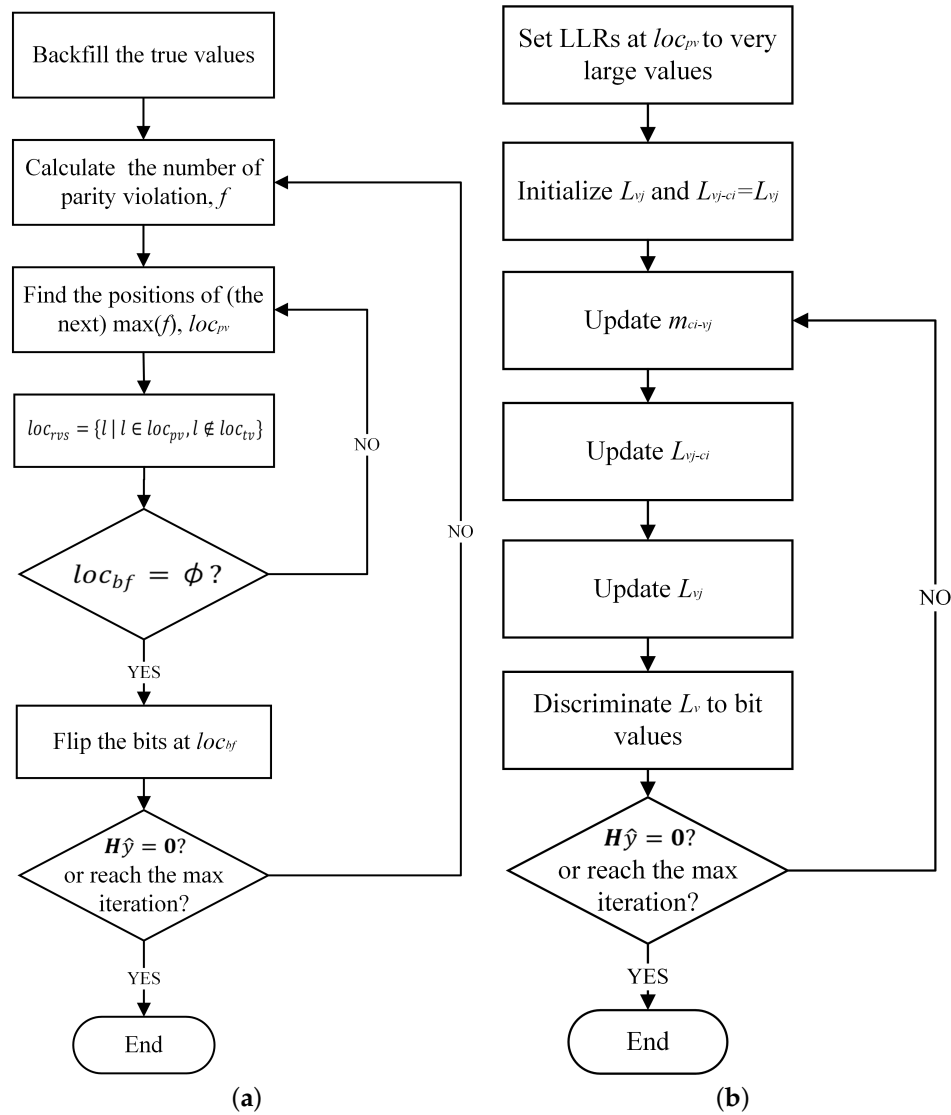


Figure 10. The enhancement strategies for (a) bit flip decoding and (b) sum-product algorithm (SPA) decoding.

Taking QC-LDPC (10,080, 8400) [22] as an instance for latency analysis, whose parity check matrix has 288 non-zero circulants, supposing that the decoder is capable to process one circulant per clock, the decoder requires 288 clock cycles to finish an iteration. As shown in Figure 13, when RBER = 0.008, the average number of iterations at $FER = 10^{-5}$ is 10 iterations with the proposed enhancement strategy whereas the iteration time is 18 without it. Supposing that 64 bits are read from the memory per clock cycle, the interleaving and de-interleaving each takes $\lceil 10,800/64 \rceil = 158$ clock cycles, and true-value sequence inserting, backfilling and removing each takes extra $\lceil 256/64 \rceil = 4$ clock cycles. Hence, $288 \times 18 - (288 \times 10 + 158 \times 2 + 4 \times 3) = 1976$ clock cycles can be saved in every decoding, which indicates that the read is sped up by 38.12%. Hence, the enhancement strategy can strengthen the error performance of SPA and thus speed up reading operation when RBER becomes high to the upper bound of ECC. From the perspective of transmission efficiency, decoding 28 frames

with the redundancy transmits almost as much amount of data as decoding 27 frames without the redundancy. The former takes $28 \times (288 \times 10 + 158 \times 2 + 4 \times 3) = 89,824$ clock cycles whereas the latter takes $27 \times 18 \times 288 = 139,968$ clock cycles, so the former saves 50,144 clock cycles when transmitting almost the same amount of data, which is equivalent to a 35.8% increase in efficiency.

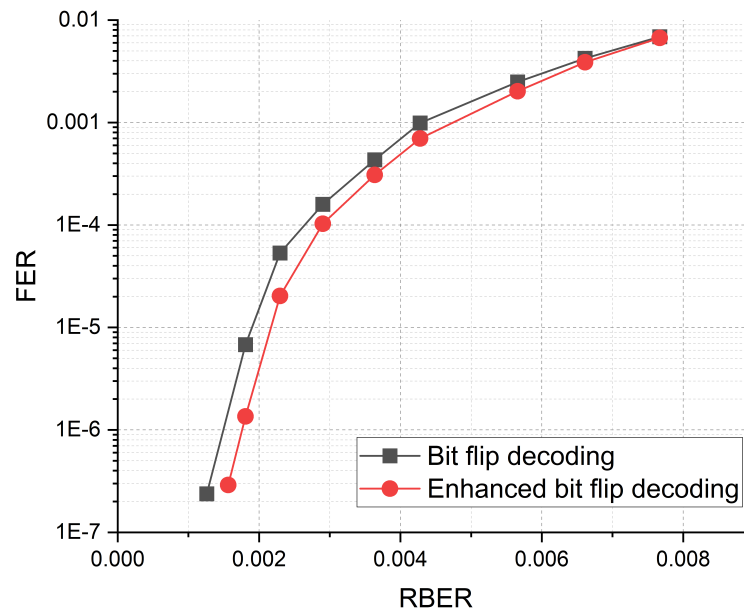


Figure 11. Bit flip decoding in maximum 10 iterations with or without the enhancement strategy, where a (10,080, 8400) QC-LDPC code is considered. True value bits are not counted in FER calculation.

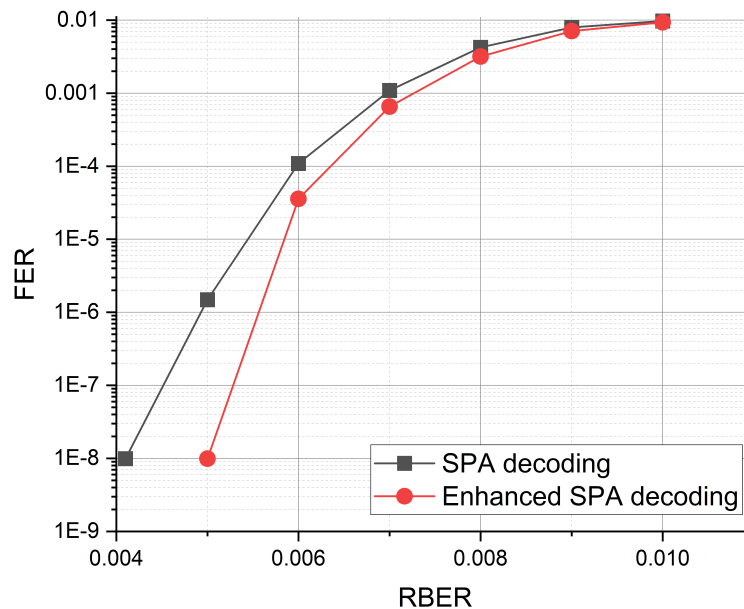


Figure 12. SPA decoding in maximum 10 iterations with or without the enhancement strategy, where a (10,080, 8400) QC-LDPC code is considered. True value bits are not counted in FER calculation.

The difference of the iterations gets larger when RBER gets higher, especially when RBER reaches the upper bound of the LDPC code, which is 0.01 in this case. Since the maximum iterations are set much less than 70 in practice, such a frame may be judged as a decoding failure and will be switched to re-read status, which will take much more clock cycles than the soft-decision decoding. Namely, the ECC can correct more error patterns with the enhancement strategy under the same decoding status. The HSD NAND flash tends to produce high RBER, so the proposed enhancement strategy

can save many clock cycles for it and thus improve read performance. Since the flash cares more about read performance than storage space, the proposed enhancement strategy will be helpful at both lower and higher RBER, especially when RBER reaches a high value near the upper bound of the ECC. Therefore, the redundancy brought by the true-value sequence is worth the increase in the converging speed of decoding.

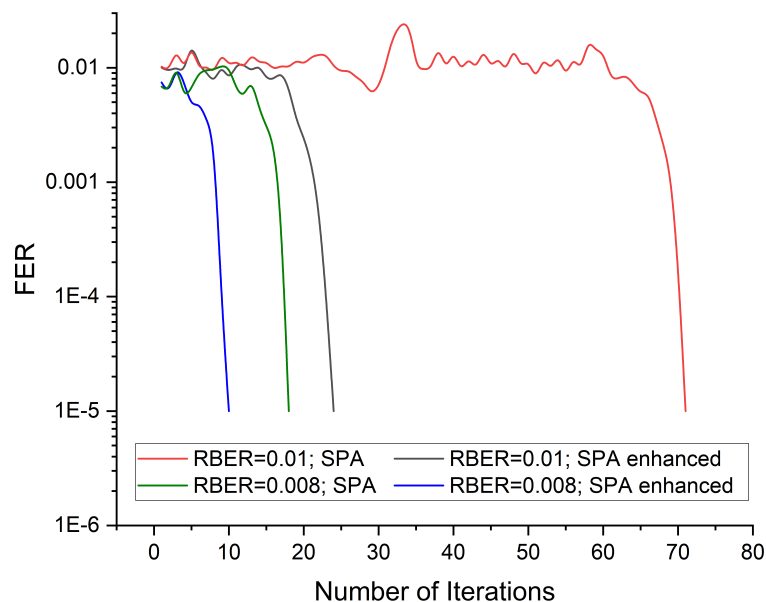


Figure 13. Frame error rate vs. number of iterations in the NAND flash memory, where the (10,800, 8400) QC-LDPC code is considered. In fact, the last iteration drops the FER to 0, but we set it to 10^{-5} for the plotting purpose, indicating that the FER less than $1/10,800$.

5. Hardware Implementation

Figure 14 represents the datapath of the proposed RBER estimation method, which is embedded in the ECC framework. When implemented with 40 nm technology library, the modules only occupy 0.039 mm^2 in total at 333 MHz clock speed at worst process corner condition, which is a very low cost for the NAND flash controller. For data programming, the input message is firstly inserted with a true-value sequence, which is for RBER estimation. The message is then encoded with the QC-LDPC code, and the codeword is interleaved for alleviating the unbalanced error distribution on pages. Afterwards, the interleaved codeword is modulated to corresponding voltage levels before it is finally programmed into the cells. For data reading, the voltage charged in cells is firstly demodulated to corresponding codes, and these codes are then deinterleaved to the original orders. Subsequently, the correct known true-value sequence is filled back into the restored codeword, and the errors are counted at the same time. The RBER estimation module will wait until collecting N frames to calculate a \overline{BER}_{est} and the collection can be done during the normal work process without the extra time required. The codeword is then decoded with the help of \overline{BER}_{est} to select a proper decoding status. Meanwhile, the enhanced strategies are applied to improve the error performance of the decoder and speed up the converge of decoding. Finally, the true-value sequence is removed before the message is output.

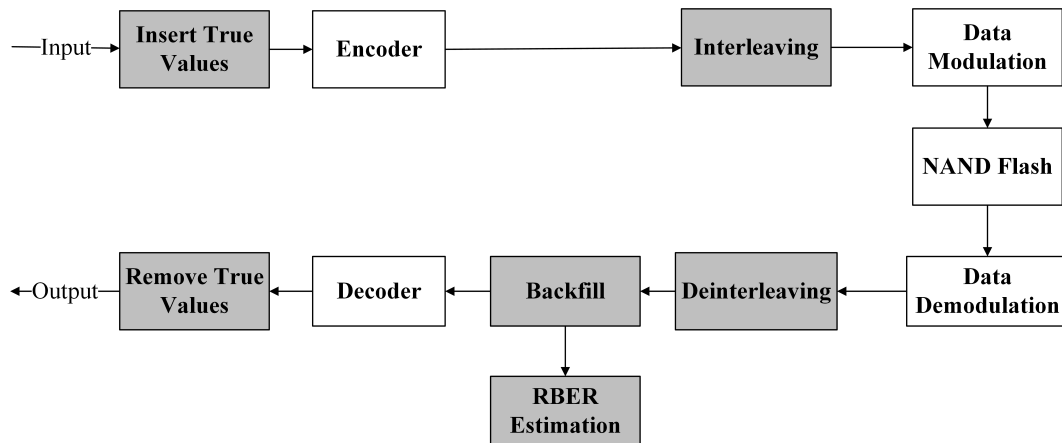


Figure 14. Datapath of the RBER estimation method embedded into the error control framework of the high storage density (HSD) NAND flash, where the grey modules are related to the proposed RBER estimation method and the white modules are related to the low-density parity-check (LDPC) error control process.

6. Conclusions

This paper proposed a fast, read-performance-improving, and low-cost RBER estimation method suitable for HSD NAND flash, including interleaving, RBER estimating and enhancement strategies for decoding. Interleaving alleviates the effect of unbalanced error distribution on pages caused by data modulation and thus improves accuracy for RBER estimation. RBER estimation is achieved with true-value data comparison, which can make estimation fast. The estimated BER is close to the actual one and they are highly linearly correlated, so the estimated BER can be sensitive to RBER change and reflect it in time. However, the redundancy brought by the true-value data reduces reading efficiency of the flash, so to solve the problem, two enhancement strategies are proposed to improve error performance of ECC in both hard-decision and soft-decision decoding. For hard-decision decoding, the improvement can raise the BER_{th} in decoding status selection so that some frames with relatively higher FER, which should have been decoded with soft-decision, can be decoded by the hard-decision decoding. Considering that hard-decision decoding has much lower computational complexity than soft-decision decoding, the enhancement will improve read performance at low RBER. For soft-decision decoding, the enhancement strategy speeds up the decoding convergence so that errors can be corrected with fewer iterations. Moreover, with the enhancement strategy, the frames with high FER close to the upper bound of ECC can be corrected within the maximum iteration times without falling in re-read status, which will take much longer than soft-decision decoding. Hence, the read performance at high RBER is also increased. The hardware complexity of the proposed RBER estimation method is very low. Therefore, the method has a high potential for long-term and frequent monitoring on RBER in HSD NAND flash.

Incidentally, more applications of the proposed method can be developed. The future work will focus on designing a reading mechanism with a neural network aided by the RBER estimation to soft sense the V_{th} shift for the SPA decoder to achieve better error correction performance.

Author Contributions: Conceptualization, K.M.; formal analysis, K.M.; funding acquisition, M.L. and H.C.; investigation, K.M. and Y.Y.; methodology, K.M.; project administration, M.L.; resources, K.M. and Y.Y.; software, K.M. and T.L.; supervision, M.L.; writing—original draft, K.M.; writing—review and editing, K.M. and M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the National Key Technologies R&D Program (Grant No. 2017YFB0405604); Key Research Program of Frontier Science, Chinese Academy of Sciences (Grant No. QYZDY-SSW-JSC004); The Basic Research Project of Shanghai Science and Technology Commission (Grant No. 16JC1400101); Beijing S&T Planning Task (Grant No. Z161100002616019).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Masuoka, F.; Asano, M.; Iwahashi, H.; Komuro, T.; Tanaka, S. A 256K flash EEPROM using triple polysilicon technology. In Proceedings of the 1985 IEEE International Solid-State Circuits Conference, New York, NY, USA, 13–15 February 1985; pp. 168–169.
2. Choi, J.-D.; Lee, J.-H.; Lee, W.-H.; Shin, K.-S.; Yim, Y.-S.; Lee, J.-D.; Shin, Y.-C.; Chang, S.-N.; Park, K.-C.; Park, J.-W. A 0.15/ μm NAND flash technology with 0.11/ μm^2 cell size for 1 Gbit flash memory. In Proceedings of the International Electron Devices Meeting 2000, Technical Digest, San Francisco, CA, USA, 10–13 December 2000; pp. 767–770.
3. Tanaka, T.; Helm, M.; Vali, T.; Ghodsi, R.; Kawai, K.; Park, J.-K.; Yamada, S.; Pan, F.; Einaga, Y.; Ghalam, A. 7.7 A 768Gb 3b/cell 3D-floating-gate NAND flash memory. In Proceedings of the 2016 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 31 January–4 February 2016; pp. 142–144.
4. Suh, K.-D.; Suh, B.-H.; Lim, Y.-H.; Kim, J.-K.; Choi, Y.-J.; Koh, Y.-N.; Lee, S.-S.; Kwon, S.-C.; Choi, B.-S.; Yum, J.-S. A 3.3 V 32 Mb NAND flash memory with incremental step pulse programming scheme. *IEEE J. Solid-State Circuit* **1995**, *30*, 1149–1156.
5. Lue, H.-T.; Hsu, T.-H.; Wang, S.-Y.; Lai, E.-K.; Hsieh, K.-Y.; Liu, R.; Lu, C.-Y. Study of incremental step pulse programming (ISPP) and STI edge effect of BE-SONOS NAND flash. In Proceedings of the 2008 IEEE International Reliability Physics Symposium, Phoenix, AZ, USA, 27 April–1 May 2008; pp. 693–694.
6. Masuoka, F. Technology trend of flash-EEPROM-Can flash-EEPROM overcome DRAM? In Proceedings of the 1992 Symposium on VLSI Technology Digest of Technical Papers, Seattle, WA, USA, 2–4 June 1992; pp. 6–9.
7. Takeuchi, K.; Tanaka, T.; Tanzawa, T. A multipage cell architecture for high-speed programming multilevel NAND flash memories. *IEEE J. Solid-State Circuit* **1998**, *33*, 1228–1238. [[CrossRef](#)]
8. Park, K.-T.; Nam, S.; Kim, D.; Kwak, P.; Lee, D.; Choi, Y.-H.; Choi, M.-H.; Kwak, D.-H.; Kim, D.-H.; Kim, M.-S. Three-dimensional 128 Gb MLC vertical NAND flash memory with 24-WL stacked layers and 50 MB/s high-speed programming. *IEEE J. Solid-State Circuit* **2014**, *50*, 204–213. [[CrossRef](#)]
9. Lee, S.; Kim, C.; Kim, M.; Joe, S.-m.; Jang, J.; Kim, S.; Lee, K.; Kim, J.; Park, J.; Lee, H.-J. A 1Tb 4b/cell 64-stacked-WL 3D NAND flash memory with 12MB/s program throughput. In Proceedings of the 2018 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 11–15 February 2018; pp. 340–342.
10. Shibata, N.; Maejima, H.; Isobe, K.; Iwasa, K.; Nakagawa, M.; Fujiu, M.; Shimizu, T.; Honma, M.; Hoshi, S.; Kawaai, T. A 70 nm 16 Gb 16-level-cell NAND flash memory. *IEEE J. Solid-State Circuit* **2008**, *43*, 929–937. [[CrossRef](#)]
11. Wang, K.L.; Du, G.; Lun, Z.Y.; Chen, W.Y.; Liu, X.Y. Modeling of program V_{th} distribution for 3-D TLC NAND flash memory. *Sci. China-Inf. Sci.* **2019**, *62*, 10. [[CrossRef](#)]
12. Cai, Y.; Haratsch, E.F.; Mutlu, O.; Mai, K. Threshold voltage distribution in MLC NAND flash memory: Characterization, analysis, and modeling. In Proceedings of the 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 18–22 March 2013; pp. 1285–1290.
13. Cai, Y.; Luo, Y.; Haratsch, E.F.; Mai, K.; Ghose, S.; Mutlu, O. Experimental Characterization, Optimization, and Recovery of Data Retention Errors in MLC NAND Flash Memory. *arXiv* **2018**, arXiv:1805.02819.
14. Cai, Y.; Luo, Y.; Ghose, S.; Mutlu, O. Read disturb errors in MLC NAND flash memory: Characterization, mitigation, and recovery. In Proceedings of the 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Rio de Janeiro, Brazil, 22–25 June 2015; pp. 438–449.
15. Liao, Y.-C.; Huang, C.-H.; Zeng, C.; Chang, H.-C. Data Analysis and Prediction for NAND Flash Decoding Status. In Proceedings of the 2017 IEEE International Memory Workshop (IMW), Monterey, CA, USA, 14–17 May 2017.
16. Nakamura, T.; Deguchi, Y.; Takeuchi, K. Adaptive Artificial Neural Network-Coupled LDPC ECC as Universal Solution for 3-D and 2-D, Charge-Trap and Floating-Gate NAND Flash Memories. *IEEE J. Solid-State Circuit* **2019**, *54*, 745–754. [[CrossRef](#)]
17. Kaynak, M.N.; Khayat, P.R.; Parthasarathy, S. On die bit error rate estimator for NAND flash memory. *IEEE Trans. Circuits Syst. II Express Briefs* **2016**, *64*, 772–776. [[CrossRef](#)]
18. Navon, A.; Sharon, E. Systems and Methods for Fast Bit Error Rate Estimation. US Patent 9,483,339, 1 November 2016.

19. Liu, S.J.; Zou, X.C. QLC NAND study and enhanced Gray coding methods for sixteen-level-based program algorithms. *Microelectron. J.* **2017**, *66*, 58–66. [[CrossRef](#)]
20. Rumberg, B.; Graham, D. Efficiency and reliability of Fowler-Nordheim tunnelling in CMOS floating-gate transistors. *Electron. Lett.* **2013**, *49*, 1484–1486. [[CrossRef](#)]
21. Taranalli, V.; Uchikawa, H.; Siegel, P.H. Channel Models for Multi-Level Cell Flash Memories Based on Empirical Error Analysis. *IEEE Trans. Commun.* **2016**, *64*, 3169–3181. [[CrossRef](#)]
22. IEEE Approved Draft Standard for Error Correction Coding of Flash Memory Using Low-Density Parity Check Codes. In *IEEE Standard SA-P1890*; IEEE: Piscataway, NJ, USA, 2017.
23. Gallager, R.G. Low-density parity-check codes. *IRE Trans. Inf. Theory* **1962**, *8*, 21–28. [[CrossRef](#)]
24. MacKay, D.J.C. Good error-correcting codes based on very sparse matrices. *IEEE Trans. Inf. Theory* **1999**, *45*, 399–431. [[CrossRef](#)]
25. Hagenauer, J.; Offer, E.; Papke, L. Iterative decoding of binary block and convolutional codes. *IEEE Trans. Inf. Theory* **1996**, *42*, 429–445. [[CrossRef](#)]
26. Kim, J.; Cho, J.; Sung, W. A high-speed layered min-sum LDPC decoder for error correction of NAND flash memories. In Proceedings of the 2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS), Seoul, Korea, 7–10 August 2011; pp. 1–4.
27. Kim, J.; Sung, W. Rate-0.96 LDPC decoding VLSI for soft-decision error correction of NAND flash memory. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2013**, *22*, 1004–1015.
28. Liao, Y.-C.; Lin, C.; Chang, H.-C.; Lin, S. A (21150, 19050) GC-LDPC Decoder for NAND Flash Applications. *IEEE Trans. Circuits Syst. I Regul. Papers* **2018**, *66*, 1219–1230. [[CrossRef](#)]
29. Li, Z.W.; Chen, L.; Zeng, L.Q.; Lin, S.; Fong, W.H. Efficient encoding of quasi-cyclic low-density parity-check codes. *IEEE Trans. Commun.* **2006**, *54*, 71–81. [[CrossRef](#)]
30. Hailes, P.; Xu, L.; Maunder, R.G.; Al-Hashimi, B.M.; Hanzo, L. A Survey of FPGA-Based LDPC Decoders. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1098–1122. [[CrossRef](#)]
31. Yaakobi, E.; Ma, J.; Grupp, L.; Siegel, P.H.; Swanson, S.; Wolf, J.K. Error characterization and coding schemes for flash memories. In Proceedings of the 2010 IEEE Globecom Workshops, Miami, FL, USA, 6–10 December 2010; pp. 1856–1860.
32. Fossorier, M.P.C. Quasi-Cyclic Low-Density Parity-Check Codes From Circulant Permutation Matrices. *IEEE Trans. Inf. Theory* **2004**, *50*, 1788–1793. [[CrossRef](#)]
33. Tanner, R.M. A recursive approach to low complexity codes. *IEEE Trans. Inf. Theory* **1981**, *27*, 533–547. [[CrossRef](#)]
34. Tanner, R.M.; Sridhara, D.; Sridharan, A.; Fuja, T.E.; Costello, D.J. LDPC block and convolutional codes based on circulant matrices. *IEEE Trans. Inf. Theory* **2004**, *50*, 2966–2984. [[CrossRef](#)]
35. Kou, Y.; Lin, S.; Fossorier, M.P.C. Low-density parity-check codes based on finite geometries: A rediscovery and new results. *IEEE Trans. Inf. Theory* **2001**, *47*, 2711–2736. [[CrossRef](#)]

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).