A Low-Complexity True Random Number Generation Scheme Using 3D-NAND Flash Memory

Ruibin Zhou[1], Jian Huang[1,*], Xianping Liu[1,2], Yuhan Wang[1], Xinrui Zhang[1], Yungen Peng[1], and Zhiyi Yu[1]
[1]School of Microelectronics Science and Technology, Sun Yat-sen University, Zhuhai, 510275, China
[2]Peng Cheng Laboratory, Shenzhen, 518055, China
*Corresponding author, E-mail: huangj573@mail.sysu.edu.cn

***Abstract*—Unpredictable true random numbers are essential in cryptographic applications and secure communications. However, implementing True Random Number Generators (TRNGs) typically requires specialized hardware devices. In this paper, we propose a low-complexity true random number extraction scheme that can be implemented in endpoint systems containing 3D-NAND flash memory chips, addressing the need for random numbers without requiring additional complex hardware. We successfully utilized the randomness of the rapid charging and discharging of shallow charge traps in 3D-NAND memory as an entropy source. The proposed approach only requires conventional user-mode erase, program, and read operations, without any special timing control. We successfully extracted random bitstream using this scheme without a post-debiasing process. We evaluated the randomness of the generated bitstream using the NIST SP 800-22 statistical test suite, and it passed all 15 tests.**

***Index Terms*—3D-NAND Flash Memory, True Random Number Generators, Endpoint System**

I. Introduction

The generation of random numbers is crucial in many areas, such as secure communications, data encryption, and scientific simulation. Implementing TRNGs can be achieved through various schemes, including the use of noise or oscillations in circuits [1], [2], chaotic system [3], nuclear radiation [4], and quantum effects [5]. Typically, specialized hardware devices are required for the TRNGs implementation. However, the high degree of specialization required by such circuits limits the development of TRNGs, people start to investigate the possibility of dual use structures. As the widespread use in electronic systems and low power consumption and ease of integration, memory devices are increasingly being used as TRNGs. Various types of memory have been reported to function as TRNGs, including volatile memory such as DRAM [6], [7] and SRAM [8], and nonvolatile memory such as Flash memory [9]–[14], Memristor [15], [16] and MRAM [17], [18].

Flash memory is considered as a promising candidate for TRNGs due to high density, low power consumption and cost-effectiveness. It typically comes in the form of commodity chips and is widely used in various applications, making it easily accessible. The state-of-the-art QLC NAND flash memory has a storage density of up to 28.5Gb/mm^2, with a maximum I/O rate of 3.2GB/s [19]. The cost per bit of data storage is tens of times lower than that of DRAM. The flash memory chips usually have very large memory arrays, where process variations can lead to significant electrical fluctuations. These fluctuations can serve as potential sources of entropy. However, commodity chips typically offer limited usable operations to end-users, making entropy extraction challenging. Therefore, there are relatively few existing solutions in this area. Currently, only a limited number of reports have demonstrated true random numbers generation in flash memory, typically utilizing read noise or operation latency [9]–[14]. To acquire the read noise, the threshold voltage (Vth) of the memory cells need to be adjusted to be near the read reference voltage. Program disturb [10], partial erase [11] or partial program [9]–[12] has been utilized for this Vth adjustment. These approaches require unconventional operations such as intended interrupt during normal erase/program to precisely adjust the Vth and maintain its stability to ensure consistent random number output, which makes practical application complex. For schemes that use latency variation as entropy, extracting random numbers is relatively complex and results in low throughput of 0.05 Kbit/s [14].

In this article, based on the stochastic nature of the charging and discharging of shallow charge traps in 3D-NAND flash memory and the back pattern effect in NAND flash, we developed a low-complexity scheme through routine erase, program, and read operations in commercially available 3D-NAND flash memory. We also studied the effectiveness and performance of this method.

II. Background

A. Shallow Charge Traps in 3D-NAND Flash Memory

It was believed that data retention errors in non-volatile memory are difficult to use for generating random numbers because these errors require a long time to manifest. However, we have discovered that the unique short-term data retention in 3D-NAND flash memory—related to the charging and discharging of shallow charge traps—can cause a large number of random errors in a short period, thereby becoming an entropy source for true random number generators.

Fig. 1(a) shows the high-level internal organization of a NAND flash memory block with a verification process of wordline 0 (WL0) after a short program. A detailed structure of the memory cell is shown in Fig. 1(b), where the charge trapping layer is sandwiched between two insulator layers. Once charges tunnel into the deep traps within the nitride layers, they can be preserved for a long time, typically for

978-3-9826741-0-0/DATE25/© 2025 EDAA

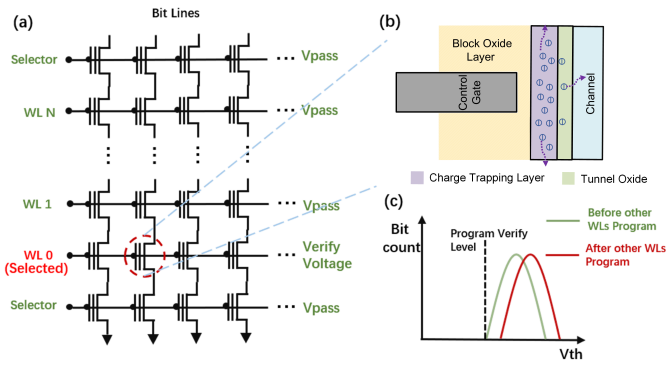

Fig. 1. (a) Schematic of the NAND flash architecture with the voltage setting during the WL0 program verify operation. (b) Schematic of a memory cell. The Charge Trapping Layer is sandwiched by the oxide layers. The blue dots indicate the charges trapped in the memory cell. The arrows indicate the possible leakage direction of the charges trapped in the shallow traps. (c) The Vth distribution of WL0 before and after other wordlines are programmed.

months, thus ensuring the non-volatility of stored information. However, Charges may also be randomly trapped by shallow charge traps within the memory cells. These unstable charges may escape from the trap sites in a very short time, leading to short-term retention issues in 3D-NAND flash [20]–[22]. Fast-detraping charges have been reported to occur in two specific locations within the memory cells. The first is within the bandgap-engineered dielectric (BE-layer), used in 3D-NAND flash memory to enhance reliability and performance. This BE-layer has defects and can trap charges and leak them quickly due to its thinness and short distance to the channel [20]. The second location is in the Silicon Nitride charge trapping layer itself, which also contains shallow trap sites where charges can be relatively easily de-trapped [22]. The status of the charges associated with shallow trap sites can serve as a potential entropy source for several reasons: 1) These trap sites predominantly result from material inconsistencies, interface anomalies, or defects related to wafer fabrication, making their locations and quantity within the cell unpredictable. 2) The charge injection and emission from a trap site are inherently stochastic [23], [24]. 3) Charges can be trapped and subsequently detrapped at various locations within the cell, and the detrapping direction and mechanism may vary [21] (Fig. 1(b)). This leads to unpredictable Vth fluctuations in individual cells and cause unpredictable read errors.

B. The Back Pattern Effect of NAND Flash Memory

The short-term retention related bit error rates are not readily apparent in commodity 3D-NAND flash memory chips because designed read reference voltage provides a sufficient margin. However, we've found that the short-term retention becomes distinctly observable through error rate characterization if we use special write data patterns to exploit the back pattern effect (BPE) of NAND flash memory. In NAND Flash memory, the block is programmed sequentially from the source-side wordline (WL) to the drain side WL. Earlier-programmed WLs, such as WL0, can experience an upward

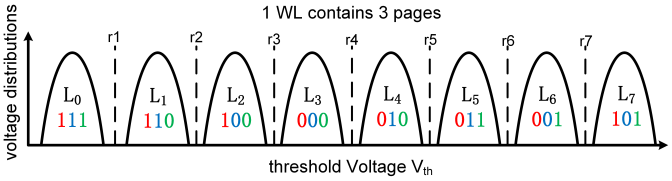

Fig. 2. The Vth distribution of a programmed TLC WL. The 8 Vth states represent 3 pages of data in one WL.

shift in the Vth after the latter WLs are programmed. This phenomenon is known as the BPE [25], [26]. The effect is associated with the series configuration of the cells and the WL program verify (PV) operation in NAND flash memory (Fig. 1(a)). When the selected cells are read or verify, the string current need to be sensed. In NAND configuration, the cells in different WLs are connected in series in one NAND string. To sense the cells in the selected WL, the cells in other WLs need to be turned on by the pass voltage. The programed cells in the unselected WLs with higher Vth can lead to an increasement of equivalent channel resistance of the NAND string comparing to erased cells under the same turn-on pass voltage. Thus, when the cells in the same string are programmed to the PV voltage before other unselected cells, its channel resistance will be lower than when other unselected cells are programmed. The sensed Vth will be higher than the target PV level when other unselected cells are programmed (Fig. 1(c)) [25], [26]. When programming a specific cell in a string to a certain voltage state, higher voltage states programmed in subsequent cells on the same string increase the influence of the BPE on that cell, leading to a larger upward shift in its threshold voltage.

III. Random Bit Extraction Mechanism

Fig. 2 shows the schematic of a WL Vth distribution of TLC NAND flash memory. In a typical block of TLC NAND flash, each WL stores three pages of data. For 3D-NAND flash that utilizes a one-pass programming method, three pages are programmed simultaneously into a single physical WL [27], [28]. This programming results in different Vth across different cells, with each cell representing three bits of data. In Fig. 2, we can see that the seven initial read reference voltages, $r1 \sim r7$, are set to distinguish the eight different threshold voltage states of TLC NAND flash. We designed a special data pattern that programs all WL in a target block to the L6 Vth state, aiming to amplify the impact of the BPE while better observing bit errors.

To generate the random number bitstream, we extract bits from the error bitmap using an even-odd method, which only utilizes the parity of the absolute physical address to determine the error location and extract random bits. The even or odd error address read from the chip determines the binary state "0" or "1" of the bitstream. This way mitigates the issue of certain locations being more error-prone due to the process limitations, particularly along the long word lines in NAND flash memory. Furthermore, considering the design and process systematic of 3D-NAND flash, two factors must be addressed:

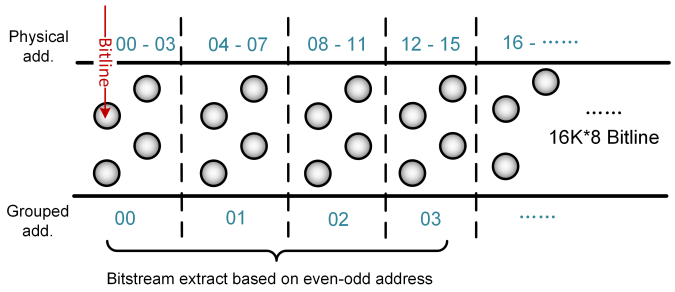

Fig. 3. Top view of a sub-block of 3D-NAND flash memory. The bitlines are grouped in sets of 4 bits to avoid system determinism interference, and the memory cells are accessed using even-odd addresses based on this grouping scheme.

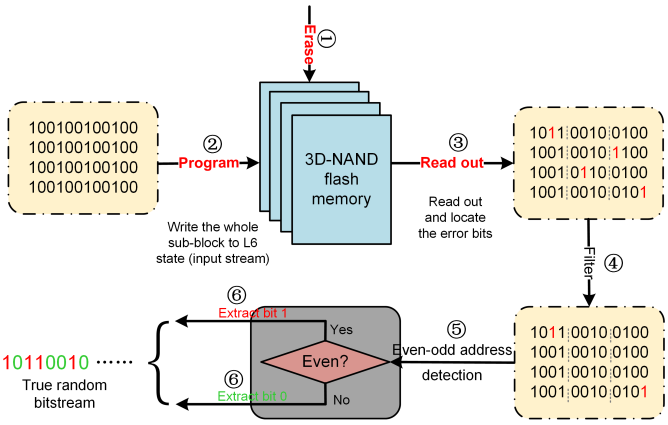

Fig. 4. Operation to generate ture random bits using 3D-NAND flash memory.

1) **Systematic Determinacy**: According to [29], [30], the design and manufacturing processes introduce systematic periodicity in the arrangement of bit lines (BLs) and channel holes, which can adversely affect randomness. For example, as shown in Fig. 3, the top view of a 3D-NAND flash memory block reveals that some channel holes are closer to the edge while others are farther away [31]. This variation may affect the shape of the memory cells and lead to differences in their ability to hold charge. To counter this, we propose combining several cells from different channels into one data unit, thereby eliminating this periodicity. By grouping 4 bits together, we only need to determine if the errors originate from the first half byte or the last half byte within one byte of data to generate the corresponding random bit.

2) **Process-Induced Weak Locations**: Weak locations in the WL, induced by manufacturing processes, may not be screened out in commercial chips. These locations can repeatedly generate errors across different erase and program cycles since we reuse the blocks to generate random numbers. To mitigate this, we selectively pick locations that exhibit only one bit error within a 32-bit span.

Fig. 4 demonstrates the detailed process of extracting true random bit streams in six steps: ① Erase the targeted block. ② Program all the pages of the targeted block with a fixed data pattern (L6 Vth state). ③ Read the page and group the read data into sets of 4 bits each, then compare it with the fixed data pattern to locate the error bits. ④ Filter the erroneous data obtained after the group comparison; if the concentration of error bits is too high within a small range (i.e., multiple error bits appear in a 32-bit section), discard that part of the data and only retain the random bits where there is exactly one error in a 32-bit section. ⑤ For groups containing erroneous bits after filtering, detect the parity address where the erroneous bit is located. ⑥ If the 4-bit data unit address is odd, extract bit 0; otherwise, extract bit 1.

Regarding the extraction method shown in Fig. 4, we should first note that bit extraction can begin immediately after the entire block is programmed, without any additional delay. This simplifies timing control and increases the throughput of the extraction process. Moreover, the variable program time of a block enhances randomness. Second, the error bits are obtained by comparing the data read out after the whole block is programmed with the designed write pattern. We do not need to compare two readout data sets before and after retention time because the charging process is also random, as mentioned in the previous section and to be studied later. This further simplifies the algorithm and saves one read operation.

IV. Test Results and Feasibility Analysis

We used a commercial NAND flash test platform to evaluate the proposed scheme and assess its feasibility. This platform supports erase, program, and read operations on different physical blocks of commercial flash memory chips. The experiments were conducted on a charge-trapping TLC 3D-NAND flash memory with 64 layers and 768 pages per block.

A. BPE and Short-Term Retention

We first tested the impact of BPE on the data error rate under different programming modes. Fig. 5(a) displays the error rate results for the first WL under various programming scenarios at different read delay time. In Case "1", only the first WL is programmed to the L6 state while the other WLs remain empty. Cases "50" and "100" represent scenarios where 50% and 100% of the targeted block, respectively, are programmed to the L6 state. In Case "100RND", the first WL is programmed to L6 while the other WLs contain pseudo-random data across all eight states. When only the first WL in the block is programmed and the others are left empty, the error rate related to the Vth upshift is low (Fig. 5(a) inset). As the number of programmed WLs increases to 50% of the block, more bit errors occur. The error bit count from WL0 is highest when the entire block is programmed to the L6 state. This is attributed to the BPE being strongest when all WLs are programmed to the L6 state. The upper tails of the L6 state cross the read reference level between the L6 and L7 states, resulting in a significant number of error bits (Fig. 5(a) inset). In NAND flash memory, the disturb from the pass gate voltage during program can also cause an upward shift in Vth of the earlier programmed WLs due to the pass voltage disturb [32].

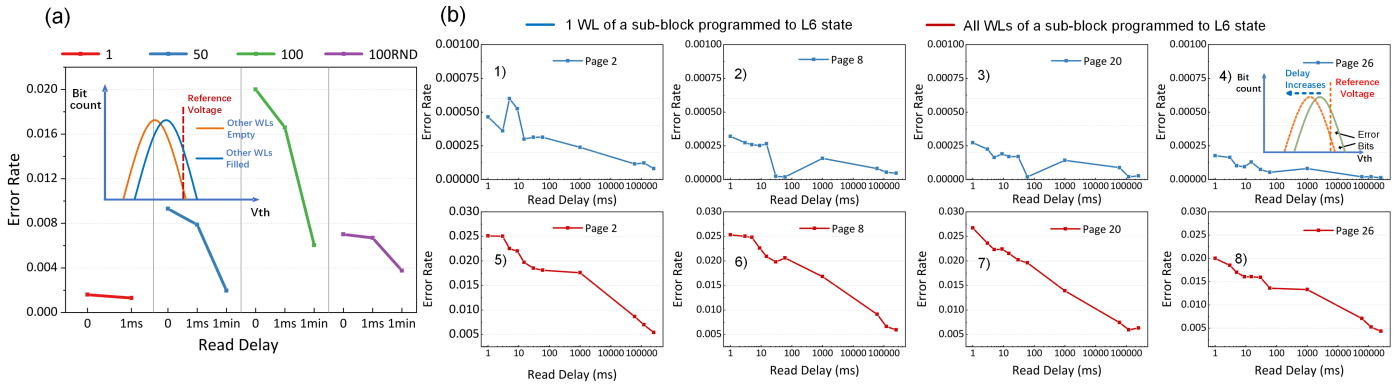

Fig. 5. (a) The error rate of the first WL for different programming scenarios at different read delay time. The number 1, 50 and 100 represents the programmed percentages of a block. "RND" indicate the block is filled with random pattern after the first WL. The inset shows the schematic of the WL0 Vth distribution before and after the rest of the WLs are programmed. (b) The short-term retention effect of four different pages (page 2, page 8, page 20, page 26) (i.e., sub figure 1) shows the error rate of page 2 (the upper page of WL0) when only WL0 is programmed to the L6 state while other WLs remain empty. sub figure 5) shows the error rate of page 2 when all WLs within a block are programmed to the L6 state), characterized by the error rate versus delay time. The inset shows the schematic of the WL0 Vth distribution right after the program and after the delay time.

To ascertain that this upward shift is mainly due to BPE, we analyzed cases "100" and "100RND". In both scenarios, WL0 experiences the same level of pass voltage disturbance after the entire block is programmed. However, the error rate for WL0 in Case "100" is significantly higher than in "100RND". This difference arises because in "100RND", other cells in the same NAND string are in mixed states, many of which have a lower Vth than L6, thus reducing the BPE. This evidence further confirms that the primary cause of the error bits is the BPE rather than program disturb.

In Fig. 5(b), we describe the short-term retention effect of four different pages (page 2, page 8, page 20, page 26) within a block under different programming modes. The blue and red lines in the Fig. 5(b) represent a single WL or all WLs in the block being programmed to the L6 state, respectively. The upper page of different selected WLs is read at different delay times after programming. A significant error rate difference can be observed between the two programming modes. When only a single WL is programmed while keeping other WLs in the block empty, the maximum error rate only reaches the order of 10^{-4}. This is attributed to the read reference voltage margin mentioned earlier, which makes the rapid charge leakage in shallow charge traps difficult to observe. When the entire bloc is programmed to the L6 state, a very high error rate is observed immediately after programming. As the waiting time increases, the error rate rapidly decreases. For the measured page, approximately 53.24% of the errors disappear on average within 10 seconds for a fresh block, which correlates with the fast charge detrapping induced Vth downshift (see Fig. 5(b) 4) inset). These experimental results demonstrate the importance of programming the entire block to the L6 state to enhance the impact of the BPE. The data also suggests that most of the error bits observed immediately after programming result from charges trapped in shallow traps.

By leveraging the BPE, the charging and discharging of shallow traps become observable as bit errors in commodity flash memory chips. This observation allows us to utilize standard erase, program, and read operations for bitstream extraction. After performing a normal erase and a full-block program, the early WLs are read to identify the error bits. Because a full-block program can take up to several hundred milliseconds, some of the shallowly-trapped charges may have already detrapped and thus the stochastic nature of the detrapping process can also enhance the randomness.

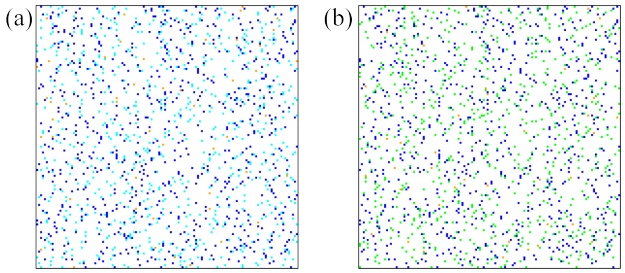

Fig. 6. Error location map of the 4-bit unit. (a) The same WL0 for 2 consecutive erase, program and read operations. Dark blue: first erase, program and read of WL0. Light blue: second erase, program and read of WL0. Orange: the overlap of the 2 operations. (b) WL0 and its neighbor WL at the first erase, program and read operation. Dark blue: first erase, program and read of WL0. Green: first erase, program and read of WL1. Orange: the overlap of the 2 WLs.

B. Independence Verification of Random Bit Generation

We also conducted independence verification on the random bits extracted repeatedly from the same WL and adjacent WLs. Fig. 6 shows the error address map for the sample pages. This map is created by sequentially folding page data into a two-dimensional image, where the first address corresponds to the top left corner and the last address to the bottom right corner. In Fig. 6(a), we overlay the error addresses of WL0 from two consecutive block erase and program (EP) cycles. The dark blue dots indicate the error addresses from the first EP cycle, while the light blue dots correspond to

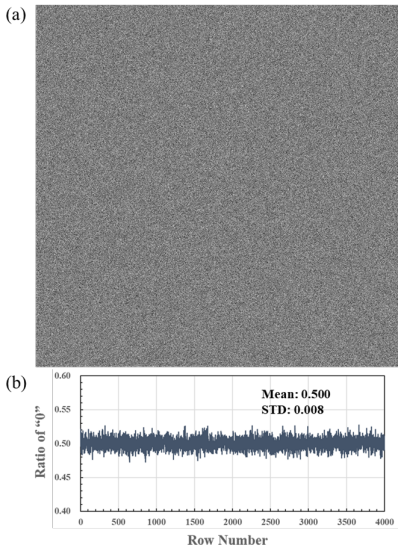

Fig. 7. (a) Two-dimensional bitmap image of a bitstream, measuring 4000 pixels by 4000 pixels. (b) The statistic of the ratio of "0" bit for the rows in the bitmap.

the second EP cycle. The orange dots denote addresses where error addresses overlap. The sparsity of overlap in the figure suggests the independence between two sequential operations on the same WL. Similarly, Fig. 6(b) demonstrates the independence between two adjacent WLs during the same block EP cycle. The dark blue dots represent the error addresses for the first WL, and the green dots for the adjacent WL, with orange dots indicating overlapping error addresses. The infrequency of overlaps observed in the figure also supports the independence between neighboring WLs at the same block program operation.

V. Evaluation

Based on the above test results, we generated the bitstream by repeatedly performing operations on a series of adjacent blocks. We executed 30 cycles of erase-program-read operations on these blocks to minimize randomness stemming from process variations across different areas of the chip. The tested device and experimental instruments used are the same as in Section IV.

Fig. 7(a) presents a two-dimensional bitmap image measuring 4000 by 4000 pixels. In this image, black pixels represent "0" bits, while white pixels represent "1" bits. The image illustrates a nearly equal distribution of "1"s and "0"s throughout the bitstream. This uniformity is further confirmed by the ratio statistics shown in Fig. 7(b). We calculated the ratio of "0" bits in each row of the bitmap. The mean ratio is 0.500 with a standard deviation of 0.008, indicating that the generated bitstream is unbiased and does not require a de-biasing process, unlike other flash memory TRNGs. This simplification enhances the random number generation process and improves the actual throughput of the TRNG [33], [34].

TABLE I
NIST 800-22 Test Suite Results of the Bitstream from the Proposed Scheme

Test	P-value	Pass Rate	Pass/Fail
Freq.	0.025193	0.917	Pass
BlockFreq. (M = 128)	0.162606	0.958	Pass
CumSum	0.002043, 0.002043	0.917, 0.917	Pass
Runs	0.437274	0.917	Pass
LongRun	0.739918	1.000	Pass
Rank	0.048716	1.000	Pass
FFT	0.350485	1.000	Pass
NonOverlapTemp (m = 9)	0.001399 (min.)	0.995 (avg.)	Pass
OverlapTemp (m = 9)	0.275709	1.000	Pass
Univ.	0.048716	1.000	Pass
AppEntropy (m = 10)	0.213309	0.958	Pass
RandExc.	0.122325 (min.)	1.000 (avg.)	Pass
RandExcVar.	0.122325 (min.)	1.000 (avg.)	Pass
Serial (m = 16)	0.048716, 0.002043	1.000, 1.000	Pass
LinComplex. (M = 500)	0.090936	1.000	Pass

The proposed scheme was assessed using the NIST SP 800-22 statistical test suite. It is a standard benchmark for randomness evaluation which comprises 15 specific randomness tests. Each test generates two key statistics: the P-value and the pass rate. A bitstream is considered to pass the test when the P-value is at least 0.0001 and the proportion meets or exceeds the threshold for each test [35]. There are multiple sub-tests in the nonoverlapping template, random excursions, and random excursions variant tests. For these tests, we report the smallest P-value from the sub-tests. We generated 24 bitstreams, each consisting of 10^6 bits. According to Table I, all bitstreams successfully passed all 15 tests, validating the randomness of the TRNG output. It is important to note that in some tests, the proportion of passing bitstreams was less than one, indicating that certain bitstreams failed specific tests. This is common for the reported TRNGs [14]–[16], as the test suite permits a certain number of failures. We observed that the bitstreams which failed typically originated from blocks with high bit error rates. This implies that these blocks may possess intrinsic defects unrelated to shallow traps, which affect the randomness of the bitstreams. Therefore, developing a rapid method to identify and exclude these defective blocks would be beneficial for future research.

We estimate the throughput of the TRNG from the following equation:

$$\text{Throughput} = \frac{\text{Bit\_Count}}{n \times t_{\text{erase}} + m \times t_{\text{program}} + l \times t_{\text{read}}} \quad (1)$$

where Bit_Count is the number of generated bits, m, n and l

represent the number of erase, program, and read operations, respectively. The latencies for erase, program and read operations are denoted as t_{erase}, $t_{program}$ and t_{read}, respectively. Here, 1.6 x 10^7 bits in the Fig. 7(a) were generated from 30 cycles of erasing and programming across 30 blocks, with 30 page reads per block for each cycle. We measured an average read latency of 67 us/page, a programming latency of 530 us/page, and a block erase latency of 4800 us/block. From equation (1), the throughput is estimated to be 43.2 Kbit/s. This throughput is based on a single-plane, single-die operation in the NAND flash tester without any parallel operations. In a real NAND flash memory system, multiple planes and dies operate in parallel, so the throughput can be significantly higher than this number.

VI. Conclusion

In this paper, we propose a low-complexity scheme to extract true random numbers from 3D-NAND flash memory. This scheme utilizes shallow trap related errors as an entropy source, with the BPE employed to uncover these errors using only normal user operations. The generated bitstreams have been successfully validated by the NIST SP 800-22 statistical test suite. This method generates random bitstreams at a speed of at least 43.2 Kbit/s. There is no need for any post de-biasing process, which further simplifies the operation. The proposed scheme provide a solution for endpoint systems containing 3D-NAND flash memory chips for convenient true random number generation.

Acknowledgment

This work was supported by the 100 Talents Program of Sun Yat-sen University (Grant No. 76220-12230040) and the Guangdong Province Key Areas R&D Plan Project (Grant No. 2023B0303030004).

References

[1] R. Brederlow, R. Prakash, C. Paulus, and R. Thewes, "A low-power true random number generator using random telegraph noise of single oxide-traps," in *2006 IEEE International Solid State Circuits Conference-Digest of Technical Papers*. IEEE, 2006, pp. 1666–1675.

[2] J. Park, B. Kim, and J.-Y. Sim, "A pvt-tolerant oscillation-collapse-based true random number generator with an odd number of inverter stages," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 10, pp. 4058–4062, 2022.

[3] P. Z. Wieczorek and K. Gołofit, "True random number generator based on flip-flop resolve time instability boosted by random chaotic source," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 4, pp. 1279–1292, 2017.

[4] K. H. Park, S. M. Park, B. G. Choi, J. B. Kim, and K. J. Son, "High rate true random number generator using beta radiation," in *AIP Conference Proceedings*, vol. 2295, no. 1. AIP Publishing, 2020.

[5] J. Cheng, S. Liang, J. Qin, J. Li, Z. Yan, X. Jia, C. Xie, and K. Peng, "Semi-device-independent quantum random number generator with a broadband squeezed state of light," *npj Quantum Information*, vol. 10, no. 1, p. 20, 2024.

[6] C. Keller, F. Gürkaynak, H. Kaeslin, and N. Felber, "Dynamic memory-based physically unclonable function for the generation of unique identifiers and true random numbers," in *2014 IEEE international symposium on circuits and systems (ISCAS)*. IEEE, 2014, pp. 2740–2743.

[7] J. S. Kim, M. Patel, H. Hassan, L. Orosa, and O. Mutlu, "D-range: Using commodity dram devices to generate true random numbers with low latency and high throughput," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2019, pp. 582–595.

[8] X. Zhang, C. Jiang, G. Dai, L. Zhong, W. Fang, K. Gu, G. Xiao, S. Ren, X. Liu, and S. Zou, "Improved performance of sram-based true random number generator by leveraging irradiation exposure," *Sensors*, vol. 20, no. 21, p. 6132, 2020.

[9] Y. Wang, W.-k. Yu, S. Wu, G. Malysa, G. E. Suh, and E. C. Kan, "Flash memory for ubiquitous hardware security functions: True random number generation and device fingerprints," in *2012 IEEE Symposium on Security and Privacy*. IEEE, 2012, pp. 33–47.

[10] B. Ray and A. Milenković, "True random number generation using read noise of flash memory cells," *IEEE transactions on electron devices*, vol. 65, no. 3, pp. 963–969, 2018.

[11] L. T. Clark, J. Adams, and K. E. Holbert, "Reliable techniques for integrated circuit identification and true random number generation using 1.5-transistor flash memory," *Integration*, vol. 65, pp. 263–272, 2019.

[12] P. Poudel, B. Ray, and A. Milenkovic, "Microcontroller trngs using perturbed states of nor flash memory cells," *IEEE Transactions on Computers*, vol. 68, no. 2, pp. 307–313, 2018.

[13] S. Larimian, M. Mahmoodi, and D. Strukov, "Lightweight integrated design of puf and trng security primitives based on eflash memory in 55-nm cmos," *IEEE Transactions on Electron Devices*, vol. 67, no. 4, pp. 1586–1592, 2020.

[14] S. Chakraborty, A. Garg, and M. Suri, "True random number generation from commodity nvm chips," *IEEE Transactions on Electron Devices*, vol. 67, no. 3, pp. 888–894, 2020.

[15] G. Kim, J. H. In, Y. S. Kim, H. Rhee, W. Park, H. Song, J. Park, and K. M. Kim, "Self-clocking fast and variation tolerant true random number generator based on a stochastic mott memristor," *Nature communications*, vol. 12, no. 1, p. 2906, 2021.

[16] H. Jiang, D. Belkin, S. E. Savel'ev, S. Lin, Z. Wang, Y. Li, S. Joshi, R. Midya, C. Li, M. Rao *et al.*, "A novel true random number generator based on a stochastic diffusive memristor," *Nature communications*, vol. 8, no. 1, p. 882, 2017.

[17] L. Rehm, M. G. Morshed, S. Misra, A. Shukla, S. Rakheja, M. Pinarbasi, A. W. Ghosh, and A. D. Kent, "Temperature-resilient random number generation with stochastic actuated magnetic tunnel junction devices," *Applied Physics Letters*, vol. 124, no. 5, 2024.

[18] C. Wang, T. Zhao, Y. Zhou, J. Hu, G. Yang, and Y. Zhang, "Spin–orbit torque true random number generator with thermal stability," *Applied Physics Letters*, vol. 124, no. 10, 2024.

[19] W. J. et al., "13.3 a 280-layer 1tb 4b/cell 3d-nand flash memory with a 28.5gb/mm2 areal density and a 3.2gb/s high-speed io rate," in *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 67, 2024, pp. 236–237.

[20] M. Kim and H. Shin, "Analysis and compact modeling of fast detrapping from bandgap-engineered tunneling oxide in 3-d nand flash memories," *IEEE Transactions on Electron Devices*, vol. 68, no. 7, pp. 3339–3345, 2021.

[21] H. Mertens, R. Ritzenthaler, A. Hikavyy, M.-S. Kim, Z. Tao, K. Wostyn, S. A. Chew, A. De Keersgieter, G. Mannaert, E. Rosseel *et al.*, "Gate-all-around mosfets based on vertically stacked horizontal si nanowires in a replacement metal gate process on bulk si substrates," in *2016 IEEE symposium on VLSI technology*. IEEE, 2016, pp. 1–2.

[22] C. Woo, S. Kim, and H. Shin, "Cell pattern dependency of charge failure mechanisms during short-term retention in 3-d nand flash memories," *IEEE Electron Device Letters*, vol. 41, no. 11, pp. 1645–1648, 2020.

[23] Y. V. Gomeniuk, R. Litovski, V. Lysenko, I. Osiyuk, and I. Tyagulski, "Current stochasticity of field emission of charge from traps in the transition layer of implanted mis structures," *Applied surface science*, vol. 59, no. 2, pp. 91–94, 1992.

[24] T. Grasser, "Stochastic charge trapping in oxides: From random telegraph noise to bias temperature instabilities," *Microelectronics Reliability*, vol. 52, no. 1, pp. 39–70, 2012.

[25] W.-C. Chen, H.-T. Lue, K.-P. Chang, Y.-H. Hsiao, C.-C. Hsieh, Y.-H. Shih, and C.-Y. Lu, "Study of the programming sequence induced back-pattern effect in split-page 3d vertical-gate (vg) nand flash," in *Proceedings of Technical Program-2014 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA)*. IEEE, 2014, pp. 1–2.

[26] G. Paolucci, M. Bertuccio, C. M. Compagnoni, S. Beltrami, A. Spinelli, A. L. Lacaita, and A. Visconti, "Impact of the array background pattern on cycling-induced threshold-voltage instabilities in nanoscale nand flash memories," *Solid-State Electronics*, vol. 113, pp. 138–143, 2015.
[27] J. K. Park and S. E. Kim, "A review of cell operation algorithm for 3d nand flash memory," *Applied Sciences*, vol. 12, no. 21, p. 10697, 2022.
[28] F. Wu, Z. Lu, Y. Zhou, X. He, Z. Tan, and C. Xie, "Ospada: One-shot programming aware data allocation policy to improve 3d nand flash read performance," in *2018 IEEE 36th International Conference on Computer Design (ICCD)*. IEEE, 2018, pp. 51–58.
[29] P. Nowakowski, M. Ray, P. Fischione, and J. Sagar, "Top-down delayering by low energy, broad-beam, argon ion milling—a solution for microelectronic device process control and failure analyses," in *2017 28th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*. IEEE, 2017, pp. 95–101.
[30] L. Crippa and R. Micheloni, "Advanced architectures for 3d nand flash memories with vertical channel," *3D Flash Memories*, pp. 167–195, 2016.
[31] R. Micheloni *et al.*, *3D Flash memories*. Springer, 2016.
[32] R. Micheloni, L. Crippa, A. Marelli, R. Micheloni, A. Marelli, and S. Commodaro, "Nand overview: from memory to systems," *Inside NAND Flash Memories*, pp. 19–53, 2010.
[33] Z. Zheng, Y. Zhang, M. Huang, Z. Chen, S. Yu, and H. Guo, "Bias-free source-independent quantum random number generator," *Optics Express*, vol. 28, no. 15, pp. 22 388–22 398, 2020.
[34] W. Wei, G. Xie, A. Dang, and H. Guo, "High-speed and bias-free optical random number generator," *IEEE Photonics Technology Letters*, vol. 24, no. 6, pp. 437–439, 2011.
[35] L. E. Bassham III, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks *et al.*, "Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications," 2010.