electronics

Article

EXPRESS: Exploiting Energy–Accuracy Tradeoffs in 3D NAND Flash Memory for Energy-Efficient Storage

Md Raquibuzzaman, Aleksandar Milenkovic and Biswajit Ray *

Department of Electrical and Computer Engineering, The University of Alabama in Huntsville, Huntsville, AL 35899, USA; mr0068@uah.edu (M.R.); milenka@uah.edu (A.M.)
* Correspondence: biswajit.ray@uah.edu

Abstract: The density and cost-effectiveness of flash memory chips continue to increase, driven by: (a) The continuous physical scaling of memory cells in a single layer; (b) The vertical stacking of multiple layers; and (c) Logical scaling through storing multiple bits of information in a single memory cell. The physical properties of flash memories impose disproportionate latency and energy expenditures to ensure the high integrity of the data during flash memory writes. This paper experimentally explores this disproportionality on state-of-the-art commercial 3D NAND flash memories and introduces EXPRESS—a technique for increasing the energy efficiency of flash memory writes by exploiting the premature termination of the flash write operations. An experimental evaluation shows that EXPRESS reduces energy expenditures by 20–50%, relative to the traditional flash writes, at the cost of a minimal loss in the data integrity (<1%). In addition, we evaluate the effects of the page-to-page variability, program–erase cycling, and data retention on the implementation of EXPRESS, and we propose enhancements to counter these effects.

Keywords: 3D NAND; flash; MLC; layer variation; energy; memory storage

Citation: Raquibuzzaman, M.; Milenkovic, A.; Ray, B. EXPRESS: Exploiting Energy–Accuracy Tradeoffs in 3D NAND Flash Memory for Energy-Efficient Storage. *Electronics* **2022**, *11*, 424. https://doi.org/10.3390/electronics11030424

Academic Editor: Seung-Ho Lim

Received: 30 December 2021
Accepted: 27 January 2022
Published: 30 January 2022

1. Introduction

Nonvolatile NAND flash memories are the basic building blocks of the data storage components found in a range of systems, including IoT and edge-computing platforms, wearable electronics, smartphones, self-driving cars, and the drones to solid-state drives (SSDs) used in personal computers and cloud computing infrastructures [1]. Energy efficiency is a key requirement for the data storage components used in emerging edge computing devices, as most of them are constrained by limited power sources [2–4]. The designers of modern flash storage systems, such as SSDs, focus exclusively on the long-term data integrity rather than on the energy efficiency. In light of the many emerging approximate computing applications, e.g., machine learning, data analytics, vision, object classification, and others as described in [5–8], where approximate and short-lived data are very common, new opportunities have arisen for developing energy-efficient approximate storage systems [9].

A typical flash-memory-based storage system consists of two discrete components: the flash storage media, with one or more flash memory devices, and a flash memory controller. Often the controller and the flash memory devices are made by different companies, and system integrators integrate these components to design storage solutions tailored for specific applications. Flash memory manufacturers comply with a chip-interfacing specification defined by the Open NAND Flash Interface (ONFI) working group [10]. This specification offers few application-agnostic storage functions, which are not tailored towards energy-efficient approximate storage applications. Thus, there remain several opportunities for the system integrators to design energy-efficient storage systems by utilizing the tradeoffs between the data accuracy and the energy efficiency that are inherent to the NAND flash memory technology.

Even though NAND flash-memory-based storage solutions require less power than other nonvolatile storage solutions, e.g., hard disk drives, they still account for a significant portion of the total energy expenditures of computing systems [11]. Several recent research proposals have emphasized the prospect of approximate storage for achieving high-energy efficiency in the emerging edge-computing applications [11–18]. To curb the energy consumption of flash memories in ultra-low-power microcontrollers, Salajegheh et al. [18] propose an energy-saving technique utilizing lower operating voltages, which jeopardize the correct memory operations. To remedy the possible loss of information, they employed: (a) Repeated in-place write operations; (b) Multiple-places write operations; or (c) The RS-Berger coding of data. They report energy savings for in-place write operations of up to 34% on lower-end microcontrollers. Similarly, a study by Tseng et al. [12] shows that up to 45% of the energy consumed could be saved using dynamic voltage scaling on the basis of the flash operations being performed. Sampson et al. [14] propose an approximate storage technique in solid-state memories by relaxing the threshold voltage margins between different memory states during the write operations by using varying program pulse widths. Through detailed simulation, they show that their proposal would make the memory write 1.7 times faster. However, the implementation of their method on the common-off-the-shelf (COTS) NAND chips requires privileged commands that are not available in the ONFI command set. Li et al. [16] propose leveraging the approximate data in the NAND flash memory to improve the read performance and enhance the reliability of regular data. Papirla et al. [19] find that the energy required by flash write operations heavily depends on data patterns. Thus, they propose an encoding scheme that minimizes the frequency of power-hungry bit patterns in codewords (‘10’ and ‘01’), reducing the total energy of the flash write operations by up to 34%. Nath et al. [20] propose a lazy amnesic-compression-based technique for storing data in flash memories. The required energy for the flash write operations is reduced by using a lossy compression; the compression ratio is adjusted on the basis of the age of the data. Mathur et al. introduce Capsule [11], a log-structured object storage system for flash memories that supports the fine-grain allocation of space for storage objects, such as streams, files, arrays, and queues. Poudel and Milenkovic [13] introduce a technique that reduces the time and energy consumed by the critical flash operations in embedded NOR memories by introducing partial or aborted flash operations.

Though these techniques demonstrate significant potential for reducing the total energy consumed, they often introduce extra overhead in time, compute resources, and/or memory space [11–18]. Moreover, they usually consider lower-density flash memories, e.g., the NOR flash memories used in low-end embedded systems [13,18]. Although using partial write operations is suggested by Sampson et al. [14] as a way to increase the energy efficiency of SSDs, its effectiveness is evaluated using a simulation-based environment only, without taking into account the physical properties of COTS flash memory chips. Consequently, the effectiveness of this approach on the COTS flash memory chips remains unknown. In addition, we are not aware of any study that considers the now dominant three-dimensional (3D) NAND flash memory technology and the unique challenges it presents. For example, the timing and data integrity parameters are often layer-dependent. Thus, we believe there is a need to explore the energy efficiency of the now dominant 3D NAND flash memories, and to experimentally evaluate the effectiveness of the techniques for improving their energy efficiency.

The complex organization of NAND flash memories and their physical properties demand disproportionate latency and energy expenditures in order to ensure high data integrity when writing data into the flash memories. This paper experimentally explores this disproportionality on state-of-the-art commercial 3D NAND flash memories and introduces EXPRESS—a technique for increasing the energy efficiency of flash memory writes. EXPRESS utilizes partial program operations, thereby exploiting the disproportionality between the latency and energy expenditures on one side, and the data accuracy on the other side. The proposed method can be implemented in the storage controller’s firmware,

without requiring any privileged flash operations or changes in the system design. An experimental evaluation shows that EXPRESS reduces energy expenditures by 20–50%, relative to the traditional flash writes, at the cost of minimal loss in the data integrity (<1%). In addition, the paper experimentally explores the impact of the page-to-page variability and the program–erase cycling on the implementation of EXPRESS, and it offers strategies to cope with these undesired effects. Compared to the existing techniques, EXPRESS offers the following advantages: (a) It can be applied to both 2D and 3D flash memories; (b) It does not require any privileged operations; (c) It can be combined with, and is orthogonal to, other techniques (e.g., voltage scaling); and (d) It does not require any data preprocessing or special data encoding. Table 1 presents a comparative analysis of the major characteristics of the previously proposed related techniques and EXPRESS.

Table 1. Comparison of EXPRESS method with existing works.

Proposed Methods	Energy Saved [%]	Bit Error Rate Range [%]	Methodology	Layer Variation Consideration	Applicable to COTS Chips
Half-wits	<34	5 to 11	Experiment	×	×
Dynamic scaling	<45	2 to 19	Experiment	×	×
Data encoding	<33	Not reported	Simulation	×	×
Approximate Storage	<40	2 to 4	Simulation	×	×
EXPRESS	<52	<1%	Experiment	√	√

The following are the key contributions of the paper:

- We explore and quantify the disproportionate trade-offs between the data accuracy and the energy efficiency of flash memory program operations, using COTS 3D NAND flash memory chips. We find that more than 20% of the energy and time is spent on improving less than 1% of the bit accuracy during the memory write operations. We shed more light on this phenomenon and identify the slow memory cells belonging to the tails of the state distributions, a main reason for the disproportionate energy–accuracy tradeoffs;
- We propose a novel technique called EXPRESS, which utilizes partial write operations to increase the energy efficiency at a minimal loss of accuracy. We characterize the NAND flash operations and experimentally explore the energy–accuracy tradeoffs as a function of the partial program time. On the basis of the results of the experimental evaluation, we propose an algorithm for choosing the partial program time that strikes an optimal balance between the energy efficiency and the data accuracy;
- We perform a detailed characterization of the page-to-page variability, the program–erase cycling effects, and the data retention effects on the effectiveness of EXPRESS. We propose several countermeasures that can be adopted to properly address these variability and reliability issues.

The rest of the paper is organized as follows: Section 2 presents the background by discussing the fundamentals of 3D NAND flash memories, the flash incremental pulse programming scheme, and the flash memory interfacing; Section 3 introduces the proposed technique; Section 4 explores the effectiveness of the proposed technique when applied to 3D flash memories operating in the SLC (single-level-cell) and MLC (multilevel-cell) modes. Section 4 also discusses the challenges due to the page-to-page variability, the program–erase cycling, and the data retention issues, and it offers enhancements to EXPRESS to address these challenges. Section 5 concludes the paper.

2. Background

2.1. Fundamentals of 3D NAND Array

Traditional 2D NAND flash technology reached its fundamental scaling limits around 2015. In response, the flash memory industry has transitioned to 3D NAND flash memory

technology. Continual advances in this technology have resulted in several generations of 3D flash memory chips, each featuring an increasing number of stacked layers, from early 32-layer to contemporary 128-layer designs. These advances promise to extend the incredible growth of the bit density over the next decade [21–23].

Figure 1a shows the device structure of a 3D NAND flash memory cell. It is essentially a floating-gate metal oxide semiconductor field effect transistor (MOSFET), with a gate-all-around cylindrical channel structure. In several 3D NAND flash memory implementations, the floating-gate (FG) layer, made of conductive polysilicon, is replaced with a charge-trap (CT) nitride layer, which acts as an insulator. The FG/CT layer is electrically insulated from the transistor's terminals by the channel and gate oxide layers, and it can trap the charge, thereby holding information even when the power is turned off. The trapped negative charge on the FG/CT effectively increases the transistor's threshold voltage (V_t), relative to the case when there is no charge trapped. Thus, a flash memory cell stores information in the form of charges (electrons). The cell is in a programmed state (logic '0') if there are enough electrons on the FG/CT so that $V_t > V_{REF}$ (the transistor is off), whereas it is in an erased state (logic '1') if there are no electrons on the FG/CT so that $V_t < V_{REF}$ (the transistor is on). To change the state of a cell, two operations are performed: program and erase. These operations require high voltages on the transistor terminals and are conducted through the oxides via the Fowler–Nordheim (FN) tunneling mechanism. The program operation charges the FG/CT with electrons, whereas the erase operation removes the charges from the FG/CT. An erase operation has to be performed to change the state of the flash memory cells from the programmed state to the erased state. The program and erase operations wear out the oxide layers, thus limiting the lifetime of a flash memory cell to ~3000−100,000 program–erase cycles, depending on the type of flash memory.

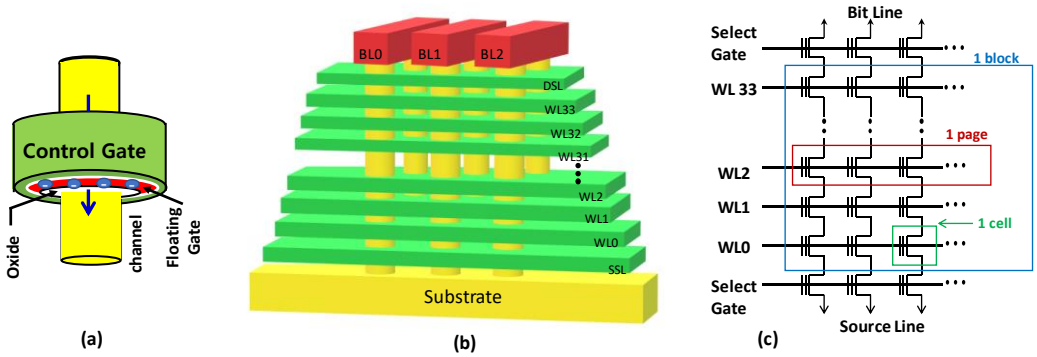

Figure 1. 3D NAND Flash Memory: (**a**) 3D flash memory cell; (**b**) physical structure of a 3D NAND flash array; (**c**) transistor-level schematic of a NAND flash memory block, which consists of 34 pages in this example.

Figure 1b shows the physical structure of the 3D NAND flash memory array. The green layers are the word lines (WL_0-WL_{33}), and the vertical pillars are the memory holes that contain the channel of the flash memory cells. Figure 1c shows the circuit diagram of the NAND flash memory array that corresponds to a single flash memory block. Each memory block consists of a fixed number of memory pages.

The cells in each memory page are electrically connected through a metal word line (WL) that acts as their control gate. Each column (or string) of cells in a block is connected to a bit line (BL). The memory *read* and *program* operations are performed at the page-level granularity, whereas the erase operations are performed at the block-level granularity. Any flash cell that is set to a logic '0' by a page program operation can only be set to a logic '1' by erasing the entire block.

A page read operation in the NAND array involves applying a read reference voltage (V_{REF}) on the selected page's WL, and then sensing the threshold voltage of the cells connected to that WL. WLs of all the other pages in the selected block are set to a high voltage (V_{PASS}), which turns on all of the flash cells from the nonselected pages. In this way,

the state of the selected page can be sensed through the bit lines. An erased cell conducts the current, and that is sensed as a logic '1', whereas a programmed cell does not conduct the current, and that is sensed as a logic '0'. The read reference voltage is set in between the erased state and the programmed state distributions to correctly identify the cell states. Traditional flash memory cells that store one bit of information are known as "single-level cells", or SLCs. The recent advances in controlling and sensing different levels of the charge on the floating gate have enabled modern flash memory cells that can store two bits of information (multilevel cell, or MLC), three bits (triple-level cell, or TLC), or even four bits (quad-level cell, or QLC).

2.2. ISPP Programming Scheme

A page program operation in the NAND array utilizes an incremental step pulse program (ISPP) scheme with multiple program cycles, as illustrated in Figure 2a. Each program cycle consists of a program pulse, followed by a verification phase. During a program pulse phase, a high voltage (~15–18 V) is applied to the corresponding WL to cause the injection of electrons into the FG/CTs of the memory cells that need to be programmed. The verification phase identifies the cells that have reached the required threshold voltage, V_t, by performing a page read operation, with a program verification voltage (V_{ref}^{PVY}) applied on the corresponding WL. Thus, the V_{ref}^{PVY} represents the minimum voltage of the program state distribution. The cells whose V_t exceed V_{ref}^{PVY} are identified as "programmed", and they are subsequently locked out from further programming using a program inhibit scheme. The following program cycle starts with an incrementally higher voltage on the WL, which increases the chances that the cells that did not switch their state in the previous cycle become programmed [24]. This sequence of the program and the verification steps continue until most of the cells that are supposed to be programmed are, indeed, programmed.

Figure 2a illustrates the different steps associated with the one-page program operation as a function of time. The steps include the high-voltage program pulse phase of the duration t_p, a relatively lower voltage verification phase of the duration t_{vfy}, and two setup time intervals—one for the program pulse of the duration, t_p^{su}, and the other for the verification phase, t_{vfy}^{su}. Assuming that all of these times remain constant across all of the program cycles, the total page program time can be expressed as follows:

$$t_{prog} = t_{init} + n \times \left(t_p^{su} + t_p + t_{vfy}^{su} + t_{vfy} \right) = t_{init} + n \times t_{pcy} \quad (1)$$

Here, $t_{pcy} = \left(t_p^{su} + t_p + t_{vfy}^{su} + t_{vfy} \right)$ represents the time required for one full program cycle, n stands for the total number of program cycles required for the page program operation, and t_{init} is the initial time required by the NAND array to verify the page status before applying a series of program cycles. Please note that Equation (1) captures the common features of the NAND page program operations. However, it may need to be adjusted depending on the specific implementation of the on-chip control logic in a particular flash memory chip.

The primary purpose of the ISPP scheme is to tighten the program state V_t distribution, relative to the initial erase state V_t distribution, which is typically wider because of the intrinsic cell-to-cell process variation. The evolution of the cell V_t distribution with the ISPP scheme is further illustrated in Figure 2b. For simplicity, we consider an SLC memory, although the same principle holds for MLC and TLC flash memories. We choose a program operation with four program cycles to illustrate the ISPP scheme. In practice, the number of program cycles could be higher. The distribution depicted with the dashed line represents the right-shifted erase state V_t distribution after each program cycle, if all the cells are programmed.

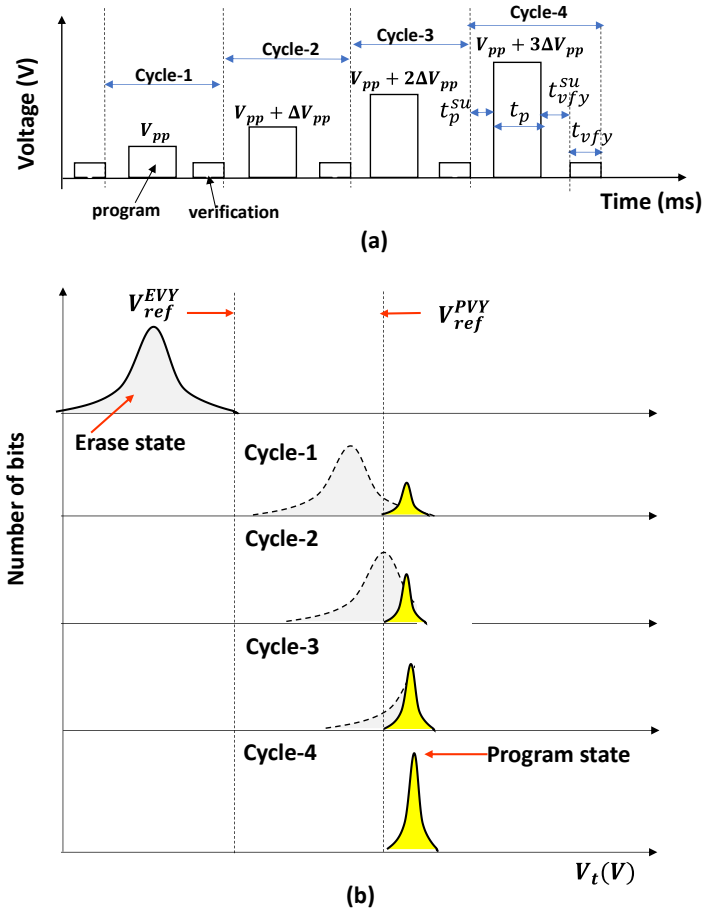

Figure 2. ISPP scheme: (**a**) an illustration of the ISPP scheme with 4 program cycles; (**b**) evolution of the V_t state for the program state over consecutive program cycles; the distribution in yellow at the bottom represents the final program state.

In practice, a certain number of cells that attain a V_t exceeding the program verification voltage are locked out of (or inhibited from) further programming cycles. Thus, the ISPP scheme tightens the cell V_t distribution by selectively providing fewer program pulses to the fast program cells, and more program pulses to the slow program cells. As a result, the final program state distribution becomes much tighter than in the erase state, as illustrated with the yellow distribution in Figure 2b. Since a tighter V_t distribution is essential for ensuring data integrity, the ISPP scheme is invariably used in all NAND flash memories. Note that the V_t distributions in Figure 2b may not follow the perfect Gaussian distribution. We used Gaussian-like distribution for illustration purposes only, and, thus, it is not a faithful illustration of actual cell V_t distributions.

The wider an erase-state V_t distribution is, the larger the number of program cycles required to complete the write operation. Note that the slow program cells may require several additional ISPP cycles. The percentage of such cells, in practice, falls well below 1% of all the flash cells in a page. Thus, the ISPP scheme entails a disproportionate energy–accuracy tradeoff, where a significant fraction of the program time, and, thus, the energy, is spent programming a tiny fraction of memory cells. The energy–accuracy tradeoff is even more skewed for 3D NAND technology, which exhibits significant cell-to-cell variations because of the poly-Si channel material and the nonuniformity in the cell dimensions caused by the reactive ion etching process [25,26]. Thus, long-tail erase-state distribution is a fundamental property of 3D NAND. Hence, the energy–accuracy tradeoffs in the ISPP programming scheme of the 3D NAND need to be evaluated carefully for energy-efficient storage applications.

2.3. Interfacing NAND Chip from the Host Controller

COTS flash memory chips use a standardized low-level interface, which was developed by the Open NAND Flash Interface (ONFI) working group, a consortium of flash memory manufacturers [10]. The ONFI specifications define: the standard physical interfaces; the chip identification mechanisms; a standard command set for reading, writing, and erasing the NAND flash; the timing requirements; and the data integrity features.

Depending on the chip package and the type of the interface, the number of bytes sent to, or received from, a device at a time can vary. In our case, both the commands and the data are carried through eight data lines (DQ0-DQ7). The control lines, CE# (Chip Enable, active low), CLE (Command Latch Enable), ALE (Address Latch Enable), RE# (Read Enable, active low), and WE# (Write Enable, active low), allow for the control of the functions and timing of the interface. As is shown in Figure 3, a command placed on the data lines by the host is written into the device's command register on the rising edge of WE# when CE# is low, ALE is low, CLE is high, and RE# is high. An address placed on the data lines by the host is written to the device's address register on the rising edge of WE# when CE# is low, ALE is high, CLE is low, and RE# is high. Data placed on the data lines by the host is written into the device's data register on the rising edge of WE# when CE# is low, ALE is low, CLE is low, and RE# is high. Data is output from the device if it is in a ready state. The data from the device's data register is output to the data lines on the falling edge of RE# when CE# is low, ALE is low, CLE# is low, and WE# is high.

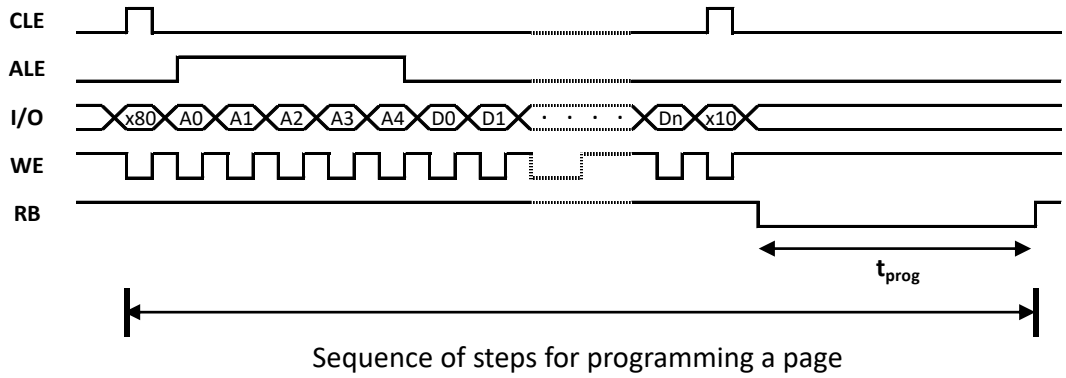

Figure 3. Timing diagram for a sequence of steps carried out by the host during the page program operation in an asynchronous interface.

Figure 3 illustrates a sequence of commands that carry out a page program operation. The operation is initiated by the host that sends the command (0x80) to the device through the data lines. Next, the host writes five address cycles (A0-A4), while keeping the ALE signal high. Next, the host controller sends the data to be written to the device's data register, byte by byte. Finally, the host sends the PAGE PROGRAM command (0x10) that initiates the write operation to the specified page of the flash memory array. During the page program operation, the device's status control pin RB (Ready/Busy#) is low, indicating that the device is currently busy. Upon completion of the program operation, the RB signal is set high. Thus, the host can determine the page program time (t_{prog}) by monitoring the state of this pin after issuing the command sequence.

3. Proposed Technique—EXPRESS

The EXPRESS technique reduces the energy consumed during the flash program operations, at the cost of a negligible loss of accuracy. It relies on a partial page program operation to counter the disproportionate energy–accuracy tradeoff inherent in the ISPP scheme.

Figure 4a illustrates the proposed EXPRESS technique. The solid black line represents the status of the RB pin during a regular page program operation. This pin goes low, indicating that the NAND array is busy for the duration of the program operation, t_{prog}. The t_{prog} value lies in the range of 300–600 µs for an SLC memory page of the chip used

in this study [27]. The program operation, however, can be terminated prematurely using a RESET command, such as the program suspend operation [28]. In this case, the state of the RB pin is illustrated with the red dashed line. The premature termination of the program operation results in a partial program operation. Although this operation may slightly increase the bit error rate (BER), it can significantly reduce the time and energy of the page program operations. The critical parameter that enables an exploration of the tradeoffs between the energy and the accuracy is the partial program time, t_{pp}. The following equation can be used to estimate the t_{pp}:

$$t_{pp} = t_{init} + \left(n - n_{skip} - 1\right) \times t_{pcy} + \left(t_p^{su} + t_p\right) = t_{prog} - \left(n_{skip} + 1\right) t_{pcy} + \left(t_p^{su} + t_p\right) \quad (2)$$

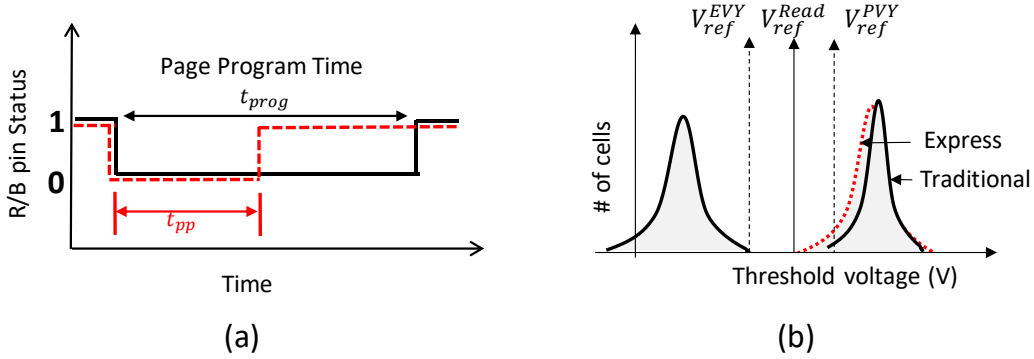

Figure 4. Nominal (solid black line) and partial-page program (dashed red line) operations: (**a**) timing diagrams; (**b**) the threshold voltage distributions for the erased and programmed states, and the corresponding reference voltages.

Here, n_{skip} is the number of program cycles that can be skipped to achieve higher energy efficiency. Note that we have not included the verification phase of the last program cycle in Equation (2), as no additional bits are programmed during the verification phase. In general, Equation (2) can be used as a guideline for finding an optimal t_{pp}, which needs to be precharacterized on the basis of the properties of the particular family of flash memory chips.

Figure 4b sheds more light on the rationale behind EXPRESS by illustrating three different reference voltages, which correspond to three different memory operations. The erase operation ensures that the threshold voltages of all the erased cells in the block are below the reference voltage, V_{ref}^{EVY}. Similarly, V_{ref}^{PVY} is the reference voltage used during the program verification phase of the page program operation. The ISPP scheme ensures that the threshold voltages of all the programmed cells are above the reference voltage, V_{ref}^{PVY}. Finally, a read reference voltage, V_{ref}^{Read}, is used to distinguish between the erase and program states of the cell during a page read operation. All NAND manufacturers keep a sufficient voltage margin between the read and program verification voltages in order to minimize read errors. However, this margin can be exploited to increase the energy efficiency in all applications where the BER is sufficiently low and it can be corrected using error-correction techniques. In addition, EXPRESS can be used even when a somewhat higher BER can be tolerated, e.g., in applications where approximate short-lived data are common. For example, if we terminate the program operation prematurely, the resulting threshold voltage distribution will be mostly above the read reference voltage, as is shown with the dashed lines in Figure 4b. The resulting distribution may have some area below the read reference voltage and that will create errors, which we are trading off for the saved energy.

Since 3D NAND flash memory cells in the erased state exhibit long tails of the threshold voltage distribution, programming these cells may require several extra program pulses. Since left-tail cells usually represent less than 1% of the total page size, a premature termination of the program operation may cause just 1% of cells to have threshold voltages below

V_{ref}^{PVY}. Interestingly, not all of these tail bits will show up as error bits during a read operation, as there is a sufficient voltage margin between the read and the program verification voltages. Thus, one can improve the energy efficiency of flash memory program operations with very little, or no, sacrifice in the bit accuracy if the partial program time is chosen appropriately. However, such partial programming may lead to increased retention loss because of the reduced reliability margin. The following section presents the experimental evaluation of the energy–accuracy tradeoffs in the state-of-the-art 3D NAND flash memory. It formulates guidelines for choosing the appropriate partial program time.

4. Experimental Evaluation

In our experimental evaluation, we use a 3D NAND flash memory chip that supports both the SLC and MLC modes of operation. Section 4.1 describes our experimental setup. Sections 4.2 and 4.3 describe the results of the experimental evaluation of EXPRESS for the SLC and MLC modes, respectively. Whereas EXPRESS promises energy savings at a negligible loss of accuracy, it is important to address any practical issues that can impact the efficacy of the proposed technique, including the page-to-page variability, the wear-out of the gate oxides, and the data retention. Hence, Section 4.4 discusses the effects of the page-to-page variability and the PE cycling on EXPRESS. Section 4.5 discusses the long-term effects of the EXPRESS mechanism on data retention. Finally, Section 4.6 puts everything together with a real-world example.

4.1. Experimental Setup

Figure 5 shows our experimental setup, which consists of a TSOP-48 socket that holds a flash memory chip, an FT2232H mini module from Future Technology Devices International (FTDI), and a workstation. The FT2232H module acts as a bridge between the workstation and the device, implementing an asynchronous 8-bit parallel interface to the device, as described in Figure 3. A software package running on the workstation executes the ONFI commands for sending data to the flash memory chip, erasing a block, writing a page, reading a page, or retrieving the data from the device. This hardware setup allows us to access raw memory bits without any error correction. We used a logic analyzer and a Digilent Analog Discovery II multifunction instrument to measure the time and capture the voltage samples from a shunt resistor connected to the power line of the TSOP socket. We performed the experimental evaluation on several 3D NAND MLC chips, with the following properties: the chip capacity is 256 Gbits; the number of blocks is 2192; each block contains 1024 pages; and each page contains 18,592 bytes of data (16,384 bytes of user data, with 2208 spare bytes reserved for storing out-of-band information, such as error correction codes). The chip was manufactured using 32-layer 3D technology.

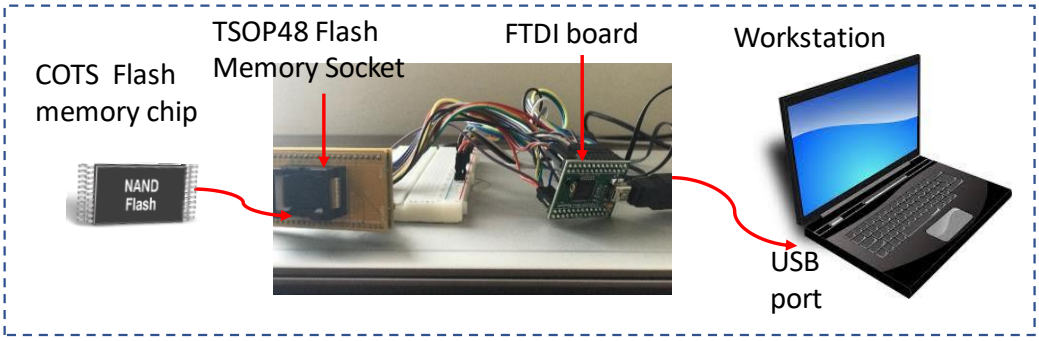

Figure 5. Experimental setup.

4.2. Evaluation of the Proposed Technique on SLC Memory

We first validate EXPRESS by configuring a NAND chip to operate in the SLC mode. An all-zero data pattern is written using partial-page program operations while varying the partial program time, t_{pp}. Later, in Section 4.4, we perform a similar experiment with

a random data pattern with an equal distribution among all the available flash cell states. Figure 6a shows the percentage of the programmed bits as a function of the partial program time. Each point in the plot represents the percentage of programmed bits collected from 10 experiments on the same page. Each partial program experiment is proceeded by a full block erase operation. Figure 6b shows the current drawn by the NAND chip during a regular page program operation. The corresponding status of the RB pin during a regular page program operation is illustrated by a red dashed line. The current drawn increases notably during the program operation relative to the current drawn in the device's idle state. The current waveform reveals two distinct profiles, which are repeated alternatively. We hypothesize that these characteristic current profiles correspond to the program (blue-shaded regions), and that they verify (red-shaded regions) the phases of the page program operation and its ISPP scheme.

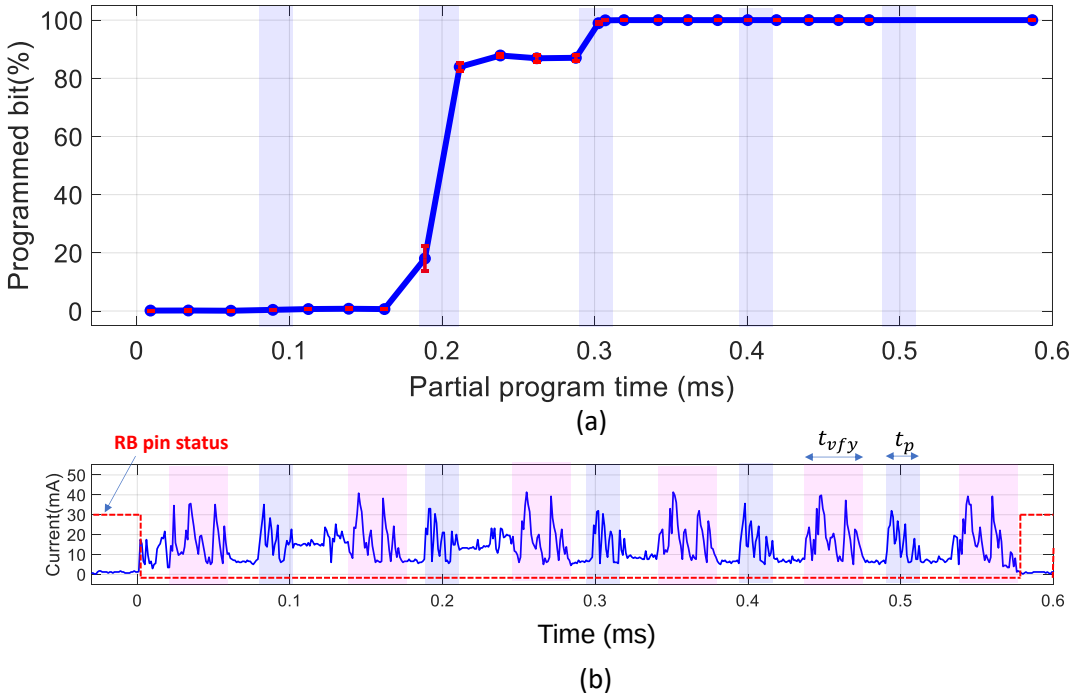

Figure 6. Effects of partial-page program operation: (**a**) percentage of programmed bits in an SLC memory page as a function of partial-page program time, t_{pp}; (**b**) the current drawn by the memory chip during a regular-page program operation.

The plot in Figure 6a shows that the percentage of programmed bits resembles a step function. The flash memory cells are programmed only during program pulses. The transition points of the percentage of programmed bits align with the program pulse phases in Figure 6b. Furthermore, the percentage of programmed bits remains constant during the verification phases. This confirms our hypothesis that the ISPP scheme is used in a page program operation, and that the characteristic waveforms correspond to the program pulses and verify the phases of the page program operation.

Furthermore, the results from Figure 6 support the following two observations:

1. Figure 6a shows that just three program cycles out of five used in a regular program operation are sufficient to achieve a bit accuracy above 99.9%. The last two program pulses are mainly used to program a tiny fraction of bits located in the lower tail of the erase V_t distribution, as illustrated in the inset of Figure 6a;
2. Figure 6b illustrates that there is periodicity in terms of the program and verification cycles, and that all program pulses and verification phases have similar duration and current profiles. Thus, Equation (2) can be used for determining a suitable partial program time. As there is no tangible advantage in the termination of the program operation in the middle of a verification or a program cycle, the optimal t_{pp} should

correspond to the end of a program pulse. The number of program pulses required to achieve the desired bit accuracy may be specific for a family of chips, the location of the page in the 3D structure, and its usage conditions. Still, all of these can be precharacterized and then used to inform a proper implementation of the partial program operations.

4.3. Evaluation of the Proposed Technique on MLC Memory

MLC flash memory cells store 2 bits of information, and, hence, there are two different types of logical pages sharing a single word line. These two bits correspond to four states of the flash memory cells, i.e., the information is encoded in the form of four threshold voltage distributions (Er-11, A-01, B-00, C-10), as illustrated in Figure 7. The most significant bit (MSB) of the logic states of all the memory cells connected to a given word line forms the MSB page. Similarly, the least significant bit (LSB) of the logic states of the memory cells from the same word line forms the logical LSB page. The LSB page programming involves raising the erase state (V_t) of certain cells to the B-state, as is shown in Figure 7. The MSB page programming is performed after the LSB page programming is finished. During MSB page programming, certain memory cells from the Er state go to the A state, and certain cells from the B state go to the C state, as is shown in Figure 7. Two read reference voltages are used to read the MSB page data, whereas only one read reference voltage is needed for reading the LSB page data.

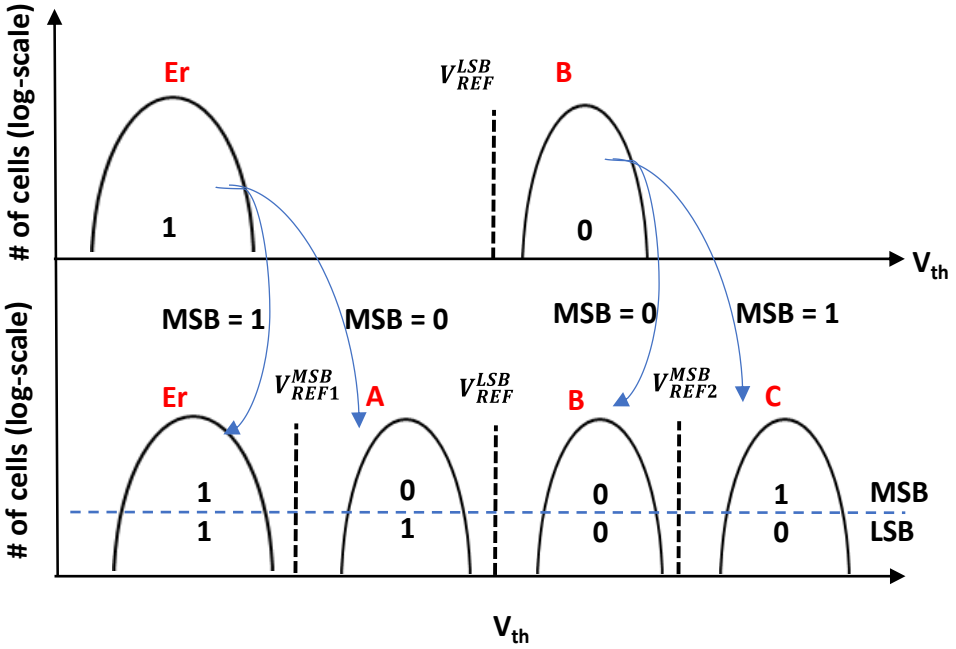

Figure 7. V_t distribution for four states in MLC mode. The top distributions show the V_t states after LSB page program, and the bottom plot shows the complete distribution after MSB page program.

Figure 8a shows the percentage of programmed bits as a function of the t_{pp} for the MSB and LSB pages in the red and blue solid lines, respectively. The experiments are conducted as follows: Logical LSB and MSB pages are used from a freshly erased block. First, we partially program an LSB page, and then the corresponding MSB page, with an all-zero data pattern. Please note that the chip used in this study, when configured in the MLC mode, by default implements data scrambling, which ensures that all four states are uniformly utilized in a physical page, regardless of the input data pattern. Thus, writing all zeros in the LSB and MSB pages does not imply that all the cells are in the B state. Therefore, the data pattern does not impact the results of our experiments, as demonstrated later in Section 4.4, where we use random data patterns. After the partial program operation, we perform the page read operation for both the LSB and MSB pages, and we determine the

percentage of programmed bits for each experiment. The programmed bit percentage for the LSB pages looks quite similar to the one observed for the SLC mode of operation. Since writing on an LSB page involves only one programmed V_t state (B state), its ISPP scheme is quite similar to the one used in the SLC mode.

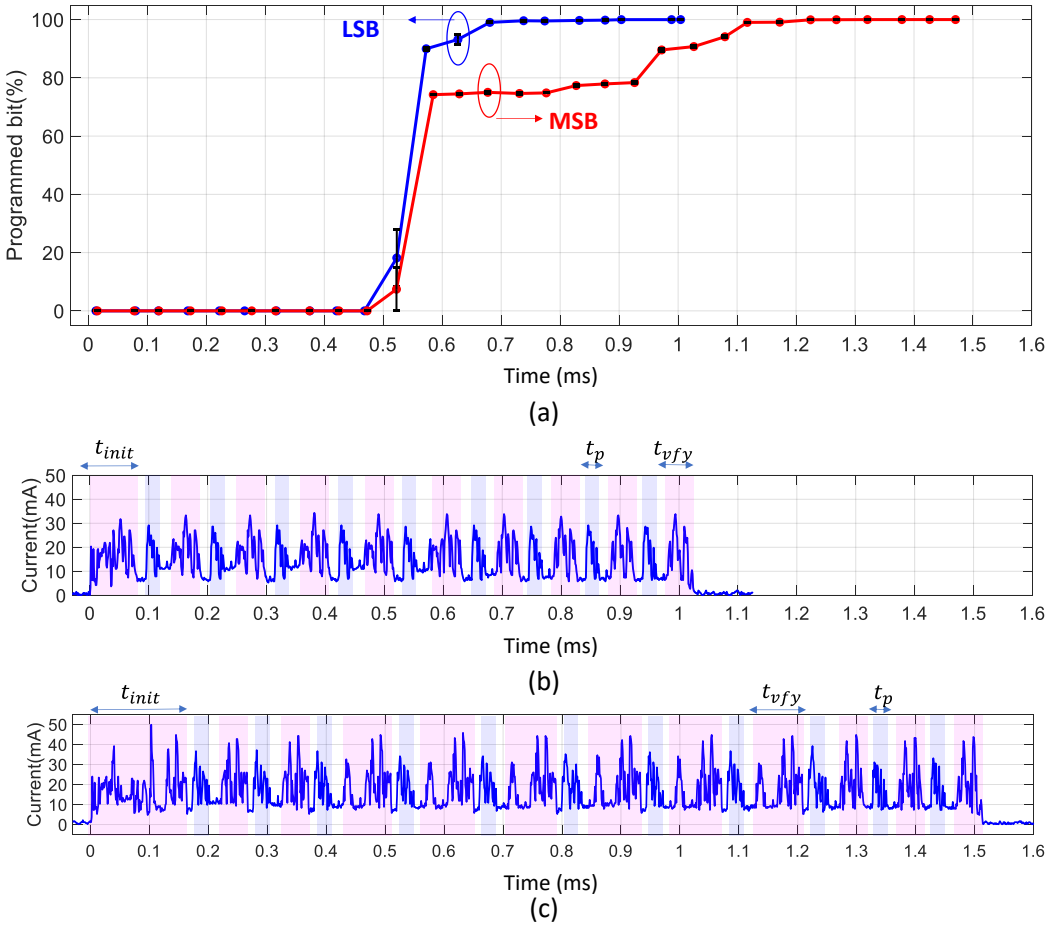

Figure 8. (**a**) percentage of programmed bits for a page in MLC mode as a function of the partial-page program time, t_{pp}. Measured current drawn during regular-page program operations in MLC mode for the (**b**) LSB page and (**c**) MSB page.

However, the programmed bit percentage for the MSB pages has distinctively different characteristics. There are two plateaus because two different V_t states, A and C, are formed during MSB programming. The first plateau corresponds to the construction of the A state, as it has a lower V_t, and is thus formed first. The second plateau corresponds to the formation of the C state. The time to complete an MSB page program operation is significantly longer than the time needed to program the corresponding LSB page. As the programming of an MSB page involves transitioning the flash cells from Er to A, and from the B to C states, it thus requires more ISPP cycles and, consequently, more time to complete a program operation, relative to its LSB counterpart. Another distinctive feature of the MSB page programming is its verification phases, which are more complex than the LSB counterparts. An LSB page verification requires only one read to verify that the cell V_t exceeds the lower bound of the B state (V_{REF}^{LSB}), whereas an MSB page verification requires two reads to check the lower bounds of both the A and C states. These hypotheses are confirmed by inspecting the current profiles, as discussed in the text below.

Figure 8b,c shows the current drawn by the chip during a page program operation for an LSB page and an MSB page, respectively. Similar to the SLC current profiles, we observe the periodic program pulses and verify the phases in the current waveform. For example, the LSB page, analyzed in Figure 8b, requires nine ISSP cycles, with the total

program time, $t_{prog}^{LSB} \approx 1000$ µs. However, the bit accuracy reaches above 99% with only seven program pulses ($t_{pp}^{LSB} \approx 750$ µs), indicating a 25% energy saving with a <1% bit accuracy loss. Programming MSB pages generally requires more time than programming LSB pages. For example, the MSB page analyzed in Figure 8c requires $t_{prog}^{MSB} \approx 1500$ µs, or 11 ISPP cycles. In addition, the verification phases in the case of MSB program operations take more time than those that take place during LSB program operations. Still, we find that partial program operations can be utilized on MSB pages, offering more than 20% in energy savings, with a negligible (<1%) bit-accuracy loss. The optimal partial program time for MSB pages is $t_{pp}^{MSB} \approx 1150$ µs.

We observe considerable page-to-page variability in the bit accuracy (error bar in Figure 8a), even though the t_{pp} was fixed. Such page-to-page variability may arise in the NAND memory because of the inherent process variations, physical organization, and the presence of program and read noise. In the next section, we elaborate further on the page-to-page timing variability and the possible countermeasures.

4.4. Effects of Page-to-Page Variability

3D NAND flash memories exhibit page-to-page variations because of the unique nature of the array geometry and the intrinsic process variations within the array. Figure 9a shows the organization of a 3D NAND memory block configured in the SLC mode. The pages in a block are organized in rows that correspond to the physical vertical layers (L_0, L_1, $\ldots L_{N-1}$) and columns that correspond to the sub-blocks (0, 1, …, $M-1$). A page number within a block can be expressed as $P_i^{L_j}$, where L_j represents the layer number, and $i = 0, 1, \ldots M-1$. The chip under evaluation has 32 layers ($N = 32$,) where each layer contains 16 logical pages ($M = 16$) of a given memory block. Thus, there are a total of $16 \times 32 = 512$ pages within a block. We performed a characterization of the page program times by sequentially programming all of the pages of a memory block using a random data pattern. Our characterization results are shown as a cumulative distribution plot in Figure 9b. The results indicate that the standard-page program time varies significantly among different pages within the same block. Since the implementation of EXPRESS requires an estimation of the t_{pp} on the basis of the nominal page program time (t_{prog}), it needs to be adapted in order to account for the page-to-page variability.

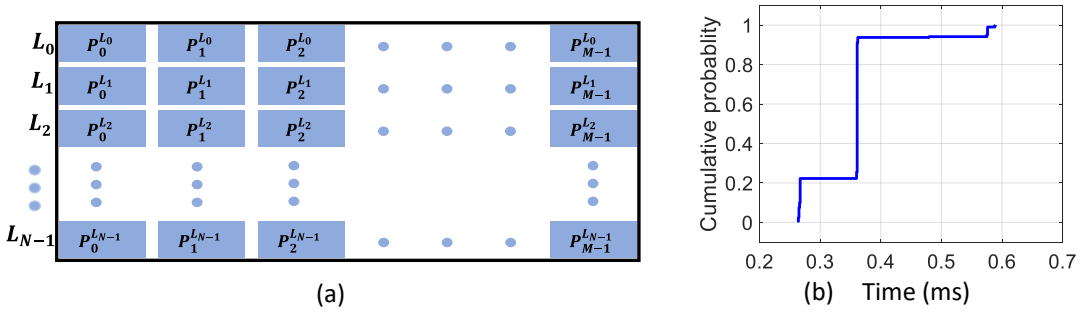

Figure 9. (**a**) organization of a 3D NAND memory block, and (**b**) cumulative distribution of the measured nominal page program time for SLC memory pages of a given block.

To understand the precise nature of the page-to-page variability, we measure each page's nominal page program time in a block in SLC mode. Figure 10 shows the results of these measurements. We can make the following two observations from these results:

a. The first page to be programmed in a given layer takes more time to complete a program operation. We classify these pages as "slow" pages, shown in blue in Figure 10;
b. The t_{prog} variability is minimal among memory pages located in the same vertical layer ($P_1^{L_j}$ *to* $P_{15}^{L_j}$) of the array. Consequently, we argue that the memory controller can learn the t_{prog} value from the page, $P_1^{L_j}$ (referred to as a "learning" page), and then apply EXPRESS when programming the remaining pages.

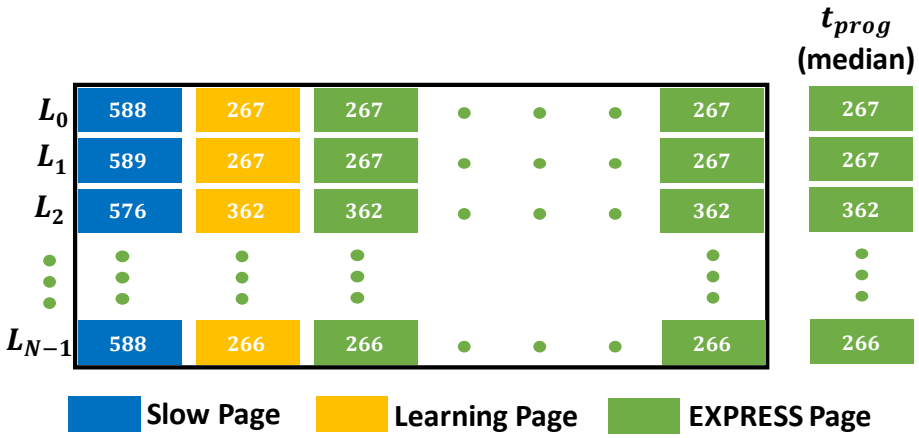

Figure 10. Classification of different SLC pages of the same memory block. The numbers represent measured t_{prog} values in µs corresponding to the page location.

To further illustrate the variability, we compute the median t_{prog} as the last column in Figure 10. We find that the median t_{prog} varies between different layers, but within the same layer, the t_{prog} remains relatively unchanged (except the first page of a layer). We exploit this observation and propose an adaptive learning algorithm to maximize the energy savings for the EXPRESS method.

To address the observed variabilities, we propose the following modification to EXPRESS. The nominal program time variation among slow pages (marked as blue boxes in Figure 10) is minimal. Consequently, the flash controller may apply EXPRESS on the slow pages by learning the corresponding t_{prog} from the first page of the block ($P_0^{L_0}$). The remaining pages of the block are classified as learning pages (yellow) and EXPRESS pages (green). The nominal page program operations are performed on the learning pages to acquire the exact t_{prog} value, and EXPRESS is applied on the remaining pages of the layer ($P_2^{L_j}$ *to* $P_{15}^{L_j}$) by estimating the corresponding t_{pp} using Equation (2).

Next, we discuss the page-to-page variability for a flash block configured in the MLC mode. Figure 11a shows the cumulative distribution of t_{prog} for the LSB and MSB pages. We find significant page-to-page variability for both the LSB (blue line) and MSB (red line) pages. LSB pages behave similarly to the SLC pages, where the first LSB page of a given layer requires significantly longer t_{prog}, compared to the other LSB pages in the same layer. These slow LSB pages constitute the upper tail (~10%) of the cumulative distribution in Figure 11a. The average t_{prog} for the MSB pages is distinctively higher than the average t_{prog} for the LSB pages. Unlike the LSB pages, the t_{prog} variability for the MSB pages is relatively small. The first MSB pages in a layer do not require a higher t_{prog} than the other MSB pages in the given layer.

Figure 11. (**a**) Cumulative distribution of program times for LSB (blue) and MSB (red) pages in a block in fresh condition; (**b**) Express algorithm for MSB pages. The numbers represent the nominal t_{prog} values in µs corresponding to the page location.

Since LSB pages behave similarly to SLC pages, the algorithm for implementing EXPRESS on LSB pages can mirror the algorithm proposed for the SLC pages, as described above. For the MSB pages, a slight modification of the algorithm is introduced by treating the first MSB page ($P_0^{L_j}$) in a given layer as the learning page. Figure 11b shows the EXPRESS algorithm for MSB pages where the t_{prog} is learned from the first MSB page of a given layer, $P_0^{L_j}$. Equation (2) is used to estimate the t_{pp} from the corresponding t_{prog}, and that value is applied to the remaining $P_n^{L_j}$ (n =1, 3, . . . , 15) pages. Next, we will discuss the energy benefits obtained with these algorithms when implemented in the 3D NAND chip under evaluation.

The adaptive learning algorithm for EXPRESS widens the opportunity window for performance and energy enhancement. Table 2 summarizes the measured t_{prog} (or nominal program time of a page), and the corresponding optimal t_{pp}, for pages in both the SLC and MLC configurations. The table also quantifies the effectiveness of EXPRESS by reporting the bit error rate and the average percentage of energy saved. The results are broken down on the basis of the page types, as discussed above. We calculate the number of program loops that can be skipped for EXPRESS to acquire an acceptable accuracy loss (<1%) for each page type. We find an optimal value of the parameter, n_{skip}, in Equation (2): $n_{skip} = 1$ or 2 for SLC pages, depending on their type, and $n_{skip} = 2$ for MLC pages. For higher values of the n_{skip}, the BER in the written data is found to be more than 1%. However, the n_{skip} needs to be precharacterized for each class of chips for optimal EXPRESS implementation. Note that the table is prepared on the basis of data collected from 1024 pages of an MLC flash block, and from 512 pages of an SLC flash block. We find that EXPRESS can save an average of 20 to 50% of the write energy, depending on the page type, whereas the exact figure for energy savings may differ for flash memory chips that have a different organization, or that are manufactured in different technology nodes. The proposed technique applies to all of them because it exploits the accuracy–energy disproportionality that is common for all modern flash memory chips.

Table 2. EXPRESS characterization for NAND flash block in fresh condition.

Storage Mode	Page Type	t_{prog}(µs)	n	t_{pp}(µs)	n_{skip}	Average BER [%]	Average Energy Saved [%]
SLC	Slow	560	5	300	2	5×10^{-1}	46.48
	Nominal	320	3	220	1	0	31.43
LSB	Slow	1022	9	760	2	5×10^{-2}	25.67
	Nominal	460	4	220	2	8×10^{-2}	52.15
MSB	Nominal	1470	11	1150	2	5×10^{-1}	21.76

4.5. Effects of Program–Erase Cycling on EXPRESS

NAND flash memory exhibits limited endurance, which is typically specified by the maximum number of program–erase operations (or PE cycles) allowed on a memory block. The number of PE cycles may impact the nominal page program time, t_{prog}, and stressed pages with a high number of PE cycles may take more time to program [29–31]. Hence, an implementation of EXPRESS needs to consider the number of PE cycles. Figure 12a shows the cumulative distribution of the nominal page program time for SLC pages in a fresh flash memory block, and in a memory block that has been exposed to 10,000 PE cycles. Similarly, Figure 12b shows the cumulative distributions of the nominal-page program times for the LSB and MSB pages in the MLC mode, for a fresh block and a block exposed to 5000 PE cycles. We find that the average t_{prog} increases with PE cycling in the MLC mode, whereas a minimal change is observed in the SLC mode.

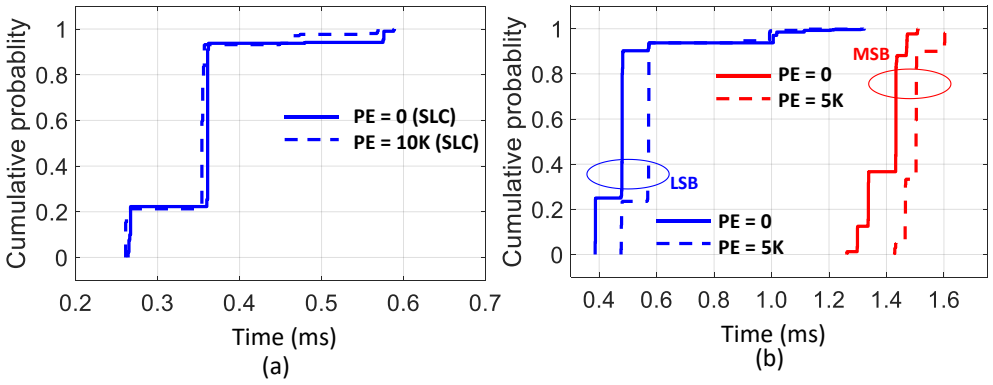

Figure 12. (**a**) Cumulative distributions of page program times for SLC pages in fresh (solid blue line) and 10K-PE-cycle (dashed blue line) conditions. (**b**) Cumulative distribution of page program time for LSB pages in fresh (solid blue line) and 5K-PE-cycle (blue dashed line) conditions, and MSB pages in fresh (solid red line) and 5K-PE-cycle (dashed red line) conditions.

Even though the average t_{prog} increases with an increase in the number of PE cycles in the MLC mode, the intralayer and interlayer t_{prog} variations remain unchanged relative to the fresh memory blocks. Specifically, our observations (a) and (b) of Section 4.4 remain true, even on stressed memory blocks. Therefore, the algorithm proposed in Section 4.4 can be used unchanged because EXPRESS learns the t_{prog} from the learning page, regardless of the PE cycles.

Table 3 summarizes the updated t_{prog} and the corresponding t_{pp} on the PE-cycled memory blocks. We find that, for 10K PE cycles in the SLC memory block, the optimal value for $n_{skip} = 1$. Higher n_{skip} values cause very high BERs in the written data. With $n_{skip} = 1$ in the SLC mode, we find that EXPRESS saves ~30% of the write energy for nominal SLC pages. Similarly, in the MLC mode operation, we find that the optimal $n_{skip} = 2$, which ensures that the BER < 1%. Thus, the energy savings are found to be ~16% for the MSB pages, and ~46% for the LSB pages. Since the t_{prog} values for the MSB pages are longer compared to the LSB pages, the percentage of energy savings is lower for the MSB pages for the same n_{skip} value.

Table 3. EXPRESS characterization for NAND flash block after PE cycles.

Storage Mode	Page Type	t_{prog}(µs)	n	t_{pp}(µs)	n_{skip}	Average BER [%]	Average Energy Saved [%]
SLC-10K	Slow	467	4	300	1	9×10^{-4}	35.78
	Nominal	320	3	220	1	5×10^{-5}	31.29
LSB-5K	Slow	1001	9	760	2	2×10^{-1}	24.12
	Nominal	561	5	300	2	1×10^{-2}	46.56
MSB-5K	Nominal	1600	12	1330	2	5×10^{-1}	16.87

4.6. Data Retention Effects

Data retention is an essential consideration for nonvolatile flash memories. The charge stored on the FG/CT of the flash cells tends to leak out through the tunnel oxides at room temperature, lowering the cell threshold voltage over a period of time [31,32]. Hence, flash memory manufacturers keep wider voltage margins between the program V_t and the read reference voltage in order to guarantee long-term data retention (~10 years for many products). Since EXPRESS trades off the voltage margin for improved energy efficiency, it is important to characterize the data retention time.

Figure 13 summarizes the results of an experiment that explores the effects of EXPRESS on the data retention for both the SLC and MLC modes of operation on PE-cycled blocks. It shows the bit error rate of the data written by EXPRESS (red bars), and the data written by the nominal program operation (blue bars). To accelerate the retention loss, we bake the chip at a higher temperature (120 °C) for 1, 2, or 3 h. Using the acceleration-factor-based

calculation, we find that the 3 h of baking time corresponds to 5 years at room temperature, assuming the activation energy for the charge loss in the 3D NAND as $E_A = 1$ eV [33]. The results in Figure 13 show that the BERs for the EXPRESS write increase relative to the traditional programming, after the accelerated retention test. The temporary read error is a new reliability issue in 3D NAND Flash [34,35]. It is not considered in this case, as the BERs are <1% for all the types of pages and, hence, can be corrected using standard error-correction techniques [36–38].

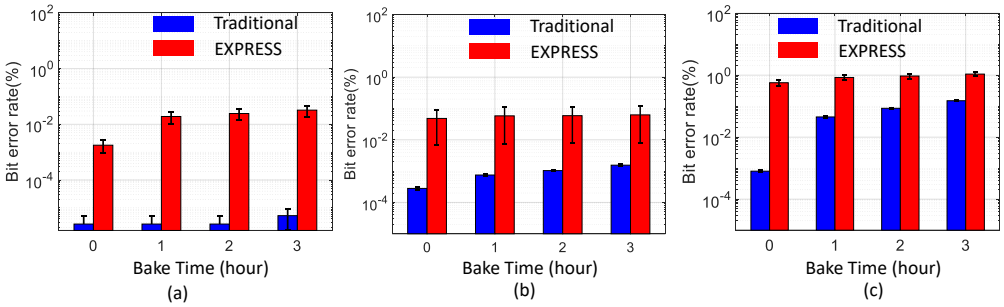

Figure 13. BER as a function of retention loss for traditional programming (blue) and proposed partial programming (red) for (**a**) SLC, (**b**) LSB, and (**c**) MSB pages.

5. Validity of the Proposed Technique for Arbitrary Image Data

In this section, we verify that EXPRESS is applicable on any data pattern, with similar results. The chip under evaluation uses an internal data randomizer that randomizes the user data before writing them in the NAND array. The goal of such data randomization is to ensure the memory reliability by utilizing all four analog V_t states. In the absence of the data randomizer, all-zero data on both the LSB and MSB pages would lead to all cells being programmed into the B state. Because of data randomization, the exact cell V_t state will be decided by the randomization key, which will ensure an even distribution of V_t states among the memory cells. Even distribution is beneficial to improving the cell endurance and reliability. Thus, randomization is an integral feature in state-of-the-art NAND flash chips [39].

In order to demonstrate EXPRESS for any arbitrary data, we write an Einstein image. Figure 14 summarizes the evaluation results for both the SLC and MLC modes of operation. We observe the same trend that we observed in Sections 4.2 and 4.3. The BER starts from 40% because the chosen image has 40% of the cells in the erase state at the beginning. Similar to earlier results, the percentage of the programmed bit exceeds 99%, with $n_{skip} = 2$. Nevertheless, it will be interesting to study the performance gain with the EXPRESS method when it is used for error-tolerant image classification applications using neuromorphic computing systems, as demonstrated in the previous works [40–42].

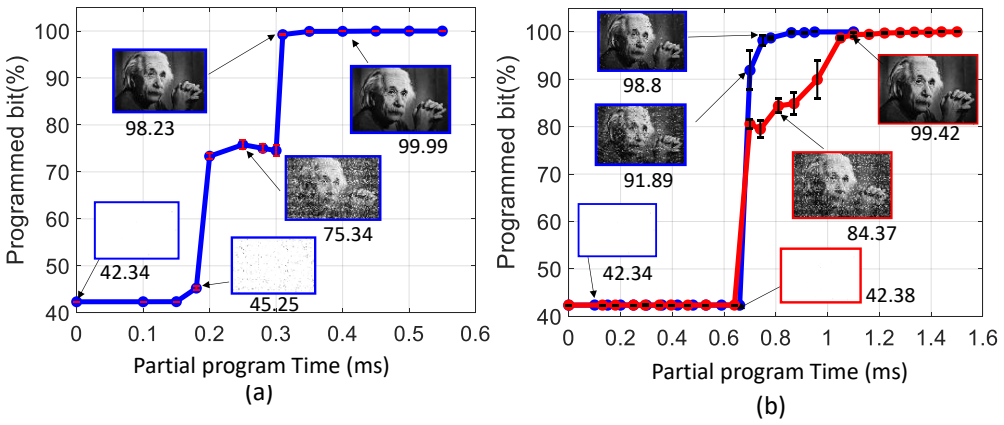

Figure 14. Percentage of programmed bits as a function of partial-page program time for the image data on (**a**) SLC, (**b**) LSB (blue), and MSB (red) pages.

6. Conclusions

In this paper, we experimentally demonstrate energy–accuracy disproportionality in 3D NAND flash memory chips. We propose EXPRESS, a new method for improving the energy efficiency of NAND write operations using a partial programming technique. We demonstrate EXPRESS on a 32-layer 3D NAND memory, operating it in both the SLC and MLC modes. We propose an adaptive algorithm for EXPRESS, considering the effects of the page-to-page variability, PE cycling, and data retention. We find that energy savings in the range of 20 to 50% are achievable, depending on the page type, at the cost of less than a 1% loss in accuracy with EXPRESS. We also find that the retention loss with EXPRESS is slightly higher than the traditional write operation. The accelerated retention test shows that the BER with the EXPRESS write remained below 1% for five years of retention time. We demonstrate the robustness of EXPRESS using an arbitrary image as a testing data pattern.

Author Contributions: Conceptualization, B.R. and A.M.; methodology, M.R.; software, M.R.; validation, M.R.; formal analysis, M.R.; investigation, M.R.; resources, B.R.; data curation, M.R.; writing—original draft preparation, M.R.; writing—review and editing, A.M. and B.R.; visualization, M.R.; supervision, B.R.; project administration, B.R.; funding acquisition, B.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Science Foundation under grant number 2007403.

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

References

1. Meena, J.S.; Sze, S.M.; Chand, U.; Tseng, T.-Y. Overview of emerging nonvolatile memory technologies. *Nanoscale Res. Lett.* **2014**, *9*, 526. [CrossRef] [PubMed]
2. Coughlin, T. Digital Storage and Memory. *Computer* **2022**, *55*, 20–29. [CrossRef]
3. Rashid, N.; Demirel, B.U.; Faruque, M.A.A. AHAR: Adaptive CNN for Energy-efficient Human Activity Recognition in Low-power Edge Devices. *IEEE Internet Things J.* **2022**, 1. [CrossRef]
4. Zanotti, T.; Puglisi, F.M.; Pavan, P. Reconfigurable Smart In-Memory Computing Platform Supporting Logic and Binarized Neural Networks for Low-Power Edge Devices. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2020**, *10*, 478–487. [CrossRef]
5. Han, J.; Orshansky, M. Approximate computing: An emerging paradigm for energy-efficient design. In Proceedings of the 2013 18th IEEE European Test Symposium (ETS), Avignon, France, 27–31 May 2013; pp. 1–6.
6. Venkataramani, S.; Chakradhar, S.T.; Roy, K.; Raghunathan, A. Approximate computing and the quest for computing efficiency. In Proceedings of the 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 8–12 June 2015; pp. 1–6.
7. Lee, Y.; Park, J.; Ryu, J.; Kim, Y. AxFTL: Exploiting Error Tolerance for Extending Lifetime of NAND Flash Storage. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2020**, *39*, 3239–3249. [CrossRef]
8. Deguchi, Y.; Nakamura, T.; Hayakawa, A.; Takeuchi, K. 3-D NAND Flash Value-Aware SSD: Error-Tolerant SSD Without ECCs for Image Recognition. *IEEE J. Solid-State Circuits* **2019**, *54*, 1800–1811. [CrossRef]
9. Reinsel, D.; Gantz, J.; Rydning, J. The Digitization of the World from Edge to Core. 2018. Available online: http://cloudcode.me/media/1014/idc.pdf (accessed on 29 December 2021).
10. Home—ONFI. Available online: http://www.onfi.org/ (accessed on 9 August 2020).
11. Mathur, G.; Desnoyers, P.; Ganesan, D.; Shenoy, P. Capsule: An energy-optimized object storage system for memory-constrained sensor devices. In Proceedings of the 4th international conference on Embedded networked sensor systems, Boulder Colorado, CO, USA, 31 October–3 November 2006; Association for Computing Machinery: New York, NY, USA, 2006; pp. 195–208.
12. Tseng, H.-W.; Grupp, L.M.; Swanson, S. Underpowering NAND flash: Profits and perils. In Proceedings of the 50th Annual Design Automation Conference, Austin, TX, USA, 29 May–7 June 2013; Association for Computing Machinery: Austin, TX, USA, 2013; pp. 1–6.
13. Poudel, P.; Milenković, A. Saving Time and Energy Using Partial Flash Memory Operations in Low-Power Microcontrollers. In Proceedings of the 2020 21st International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, USA, 25–26 March 2020; pp. 183–189.
14. Sampson, A.; Nelson, J.; Strauss, K.; Ceze, L. Approximate storage in solid-state memories. In Proceedings of the 2013 46th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), Davis, CA, USA, 7–11 December 2013; pp. 25–36.
15. Ranjan, A.; Venkataramani, S.; Fong, X.; Roy, K.; Raghunathan, A. Approximate storage for energy efficient spintronic memories. In Proceedings of the 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 8–12 June 2015; pp. 1–6.

16. Li, Q.; Shi, L.; Yang, J.; Zhang, Y.; Xue, C.J. Leveraging Approximate Data for Robust Flash Storage. In Proceedings of the 56th Annual Design Automation Conference 2019, Las Vegas, NV, USA, 2–6 June 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 1–6.
17. Salajegheh, M.; Wang, Y.; Jiang, A.; Learned-Miller, E.; Fu, K. Half-Wits: Software Techniques for Low-Voltage Probabilistic Storage on Microcontrollers with NOR Flash Memory. *ACM Trans. Embed. Comput. Syst.* **2013**, *12*, 91:1–91:25. [CrossRef]
18. Salajegheh, M.; Wang, Y.; Fu, K.; Jiang, A.; Learned-Miller, E. Exploiting half-wits: Smarter storage for low-power devices. In Proceedings of the 9th USENIX conference on File and stroage technologies, San Jose, CA, USA, 15–17 February 2011; USENIX Association: Berkeley, CA, USA, 2011; p. 4.
19. Papirla, V.; Chakrabarti, C. Energy-aware error control coding for Flash memories. In Proceedings of the 2009 46th ACM/IEEE Design Automation Conference, San Francisco, CA, USA, 26–31 July 2009; pp. 658–663.
20. Nath, S. Energy efficient sensor data logging with amnesic flash storage. In Proceedings of the 2009 International Conference on Information Processing in Sensor Networks, San Francisco, CA, USA, 13–16 April 2009; pp. 157–168.
21. Micheloni, R.; Aritome, S.; Crippa, L. Array Architectures for 3-D NAND Flash Memories. *Proc. IEEE* **2017**, *105*, 1634–1649. [CrossRef]
22. Resnati, D.; Goda, A.; Nicosia, G.; Miccoli, C.; Spinelli, A.S.; Compagnoni, C.M. Temperature Effects in NAND Flash Memories: A Comparison Between 2-D and 3-D Arrays. *IEEE Electron. Device Lett.* **2017**, *38*, 461–464. [CrossRef]
23. Compagnoni, C.M.; Goda, A.; Spinelli, A.S.; Feeley, P.; Lacaita, A.L.; Visconti, A. Reviewing the Evolution of the NAND Flash Technology. *Proc. IEEE* **2017**, *105*, 1609–1633. [CrossRef]
24. Takeuchi, K.; Tanaka, T.; Tanzawa, T. A multipage cell architecture for high-speed programming multilevel NAND flash memories. *IEEE J. Solid-State Circuits* **1998**, *33*, 1228–1238. [CrossRef]
25. Luo, Y.; Ghose, S.; Cai, Y.; Haratsch, E.F.; Mutlu, O. Improving 3D NAND Flash Memory Lifetime by Tolerating Early Retention Loss and Process Variation. *Proc. ACM Meas. Anal. Comput. Syst.* **2018**, *2*, 1–48. [CrossRef]
26. Goda, A. 3-D NAND Technology Achievements and Future Scaling Perspectives. *IEEE Trans. Electron. Devices* **2020**, *67*, 1373–1381. [CrossRef]
27. Micron 3D NAND Flash. Available online: https://www.micron.com/products/nand-flash (accessed on 29 December 2021).
28. Zambelli, C.; Zuolo, L.; Aldarese, A.; Scommegna, S.; Micheloni, R.; Olivo, P. Assessing the Role of Program Suspend Operation in 3D NAND Flash Based Solid State Drives. *Electronics* **2021**, *10*, 1394. [CrossRef]
29. Sakib, S.; Kumari, P.; Talukder, B.; Rahman, M.; Ray, B.; Sakib, S.; Kumari, P.; Talukder, B.M.S.B.; Rahman, M.T.; Ray, B. Non-Invasive Detection Method for Recycled Flash Memory Using Timing Characteristics. *Cryptography* **2018**, *2*, 17. [CrossRef]
30. Kumari, P.; Talukder, B.M.S.B.; Sakib, S.; Ray, B.; Rahman, M.T. Independent detection of recycled flash memory: Challenges and solutions. In Proceedings of the 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), Washington, DC, USA, 30 April–4 May 2018; pp. 89–95.
31. Grupp, L.M.; Caulfield, A.M.; Coburn, J.; Swanson, S.; Yaakobi, E.; Siegel, P.H.; Wolf, J.K. Characterizing flash memory: Anomalies, observations, and applications. In Proceedings of the 2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), New York, NY, USA, 12–16 December 2009; pp. 24–33.
32. Cai, Y.; Haratsch, E.F.; Mutlu, O.; Mai, K. Error patterns in MLC NAND flash memory: Measurement, characterization, and analysis. In Proceedings of the 2012 Design, Automation Test in Europe Conference Exhibition (DATE), Dresden, Germany, 12–16 March 2012; pp. 521–526.
33. Luo, Y.; Ghose, S.; Cai, Y.; Haratsch, E.F.; Mutlu, O. HeatWatch: Improving 3D NAND Flash Memory Device Reliability by Exploiting Self-Recovery and Temperature Awareness. In Proceedings of the 2018 IEEE International Symposium on High Performance Computer Architecture (HPCA), Vienna, Austria, 24–28 February 2018; pp. 504–517.
34. Zambelli, C.; Micheloni, R.; Scommegna, S.; Olivo, P. First Evidence of Temporary Read Errors in TLC 3D-NAND Flash Memories Exiting From an Idle State. *IEEE J. Electron. Devices Soc.* **2020**, *8*, 99–104. [CrossRef]
35. Xia, S.; Jia, X.; Jin, L.; Luo, Z.; Song, Y.; Liu, C.; Xu, F.; Li, K.; Li, H.; Li, D.; et al. Analysis and Optimization of Temporary Read Errors in 3D NAND Flash Memories. *IEEE Electron. Device Lett.* **2021**, *42*, 820–823. [CrossRef]
36. Mielke, N.; Marquart, T.; Wu, N.; Kessenich, J.; Belgal, H.; Schares, E.; Trivedi, F.; Goodness, E.; Nevill, L.R. Bit error rate in NAND Flash memories. In Proceedings of the 2008 IEEE International Reliability Physics Symposium, Phoenix, AZ, USA, 27 April–1 May 2008; pp. 9–19.
37. Chen, F.; Zhang, T.; Zhang, X. Software Support Inside and Outside Solid-State Devices for High Performance and High Efficiency. *Proc. IEEE* **2017**, *105*, 1650–1665. [CrossRef]
38. Kavcic, A.; Patapoutian, A. The Read Channel. *Proc. IEEE* **2008**, *96*, 1761–1774. [CrossRef]
39. Surendranathan, U.; Kumari, P.; Wasiolek, M.; Hattar, K.; Boykin, T.; Ray, B. Gamma Ray Induced Error Pattern Analysis for MLC 3-D NAND Flash Memories. *IEEE Trans. Nucl. Sci.* **2021**, *68*, 733–739. [CrossRef]
40. Xiao, T.P.; Bennett, C.H.; Feinberg, B.; Agarwal, S.; Marinella, M.J. Analog architectures for neural network acceleration based on non-volatile memory. *Appl. Phys. Rev.* **2020**, *7*, 031301. [CrossRef]
41. Fuller, E.J.; Gabaly, F.E.; Léonard, F.; Agarwal, S.; Plimpton, S.J.; Jacobs-Gedrim, R.B.; James, C.D.; Marinella, M.J.; Talin, A.A. Li-Ion Synaptic Transistor for Low Power Analog Computing. *Adv. Mater.* **2017**, *29*, 1604310. [CrossRef] [PubMed]
42. Nikam, R.D.; Lee, J.; Choi, W.; Banerjee, W.; Kwak, M.; Yadav, M.; Hwang, H. Ionic Sieving Through One-Atom-Thick 2D Material Enables Analog Nonvolatile Memory for Neuromorphic Computing. *Small* **2021**, *17*, 2103543. [CrossRef] [PubMed]