International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:10, No:4, 2016

The Characterisation of TLC NAND Flash Memory, Leading to a Definable Endurance/Retention Trade-Off

Sorcha Bennett, Joe Sullivan

Abstract—Triple-Level Cell (TLC) NAND Flash memory at, and below, 20nm (nanometer) is still largely unexplored by researchers, and with the ever more commonplace existence of Flash in consumer and enterprise applications there is a need for such gaps in knowledge to be filled. At the time of writing, there was little published data or literature on TLC, and more specifically reliability testing, with a further emphasis on both endurance and retention. This paper will give an introduction to NAND Flash memory, followed by an overview of the relevant current research on the reliability of Flash memory, along with the planned future work which will provide results to help characterise the reliability of TLC memory.

Keywords—TLC NAND flash memory, reliability, endurance, retention, trade-off, raw flash, patterns.

I. Introduction

UP until relatively recently spinning hard disk drives were the most common form of permanent data storage. However, this space is now being rapidly filled by Solid State Drives (SSD's), which use NAND Flash memory for storage. Flash memory is non-volatile, meaning that it does not lose data when the power source is removed. It has a complex memory cell structure, which means it can be written to, and erased, by electrical methods [1]. It was called Flash because the data could be erased very quickly - in a flash [2].

Important reliability metrics with regards to Flash memory are *endurance* and *retention*. Endurance is a measure of how many program/erase (P/E) cycles a cell can endure before failure [3]. The endurance values vary between device types and also between manufacturers. Common values for Single Level Cell (SLC) can be 100,000, for Multi-Level Cell (MLC) can be 5,000-10,000, while for Triple Level Cell (TLC) it can be as little as 500 P/E cycles.

Retention is a measure of how long a device can retain settings without being refreshed. According to the JEDEC specification [4] for Flash, these figures should be 1 year for 100% of the maximum cycle count, and 10 years for 10% of the maximum cycle count. This means that if a Flash device is cycled to 100% of its maximum P/E cycle count, then it has to keep the data for 1 year, and if it's cycled at only 10%, then it has to keep the data for 10 years.

P/E cycling creates significant endurance and retention problems which cause the eventual wearout of all Flash memory devices [1]. The physics of Flash mean that the electrical stress associated with changing state are the most common cause of threshold voltage (V_{th}) disturbances [5]. The V_{th} of a cell is the gate voltage at which it is turned on, and disturbances can occur due to such things as degradation in the tunnel oxide, cell-to-cell interference, and electron leakage. Several methods are employed to combat this wearout mechanism, including Wear Leveling and Error Correction Codes (ECCs), all of which are carried out by the Flash memory controller. This controller creates a single error free data stream from multiple NAND devices and hides the complexity of doing so from the user. It is typically comprised of a host interface and a Flash File System (FFS).

Wear Leveling is required because, without it, data may be continually updated in the same location, leaving other locations less-frequently updated, or not used at all. This can lead to specific, frequently updated blocks wearing out prematurely. To prevent this, the usage of all pages must be kept as level as possible. ECCs are used to correct read errors and are executed from the spare area of the memory. There are many types of ECC, but the most well-known are Reed-Solomon and Bose & Ray-Chaudhuri (BCH) [6]. Current generation TLC NAND devices require more even more powerful ECC, such as low-density parity-check (LDPC) codes, which have the ability to use soft information from multiple reads to help correct errors [7]–[9]. While performing read operations, ECCs are required to deal with various issues including noise, V_{th} disturbances, retention, and related errors. They are used to increase both endurance and retention of the Flash.

II. Background

There are two distinct types of Flash memory - *NOR* and *NAND*. NOR provides fast random memory read access and so, is used to store code and parameter data, because it guarantees 100% good bits [10]. Random access means the memory can be directly addressed and data can be found in any order, anywhere. As shown in Fig. 1, each cell is connected to both the bit and source line, facilitating random access. NAND is better for applications that need serial read access, whereas NOR is better when random read access is required. NAND does allow random access but data access is slower than NOR [10], due to the requirement to read an entire page of data (x bytes) at a time. Random write has been shown to be as fast on raw NAND Flash as serial write access, but slower on Solid State Devices (SSDs) [11].

S. Bennett is with the Department of Electrical and Electronic Engineering, Limerick Institute of Technology, Limerick, Ireland (e-mail: sorcha.bennett@lit.ie).
Dr. J. Sullivan is with Limerick Institute of Technology, Ireland.

Serial access facilitates data extraction by passing the data through the rest of the cells in the string, which are put into pass mode, by turning all the cells on. This allows access to the required cell. All cells on a Word Line must be read together and form a page of data, as shown in Fig. 2. This diagram shows that each bit line is shared by a string of cells, therefore allowing serial access. NAND is denser and cheaper than NOR, so has taken over for use in data storage, memory cards, mobile phones and SSDs - where the cost per bit is critical. This fact, along with increased demand for smaller devices, has caused the NAND Flash market to grow to over $8.5 billion in just the third quarter of 2014 alone [12], with TLC expected to account for more than 65% of the market by 2018 [13].

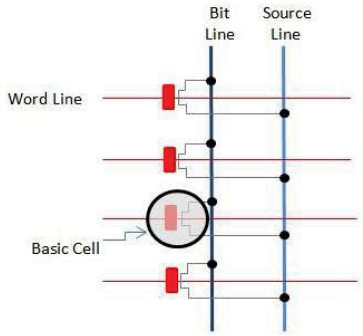

Fig. 1 NOR Flash Architecture

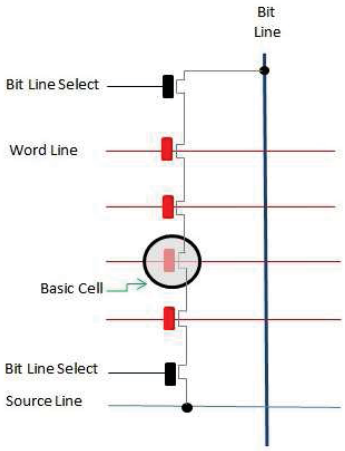

Fig. 2 NAND Flash Architecture

Both NOR and NAND are based on a Floating Gate (FG) technology consisting of a MOS (Metal Oxide Silicon) Field Effect Transistor or MOSFET. The MOS structure has three layers - the Metal layer is the control gate, the Oxide layer holds the floating gate, and the Silicon layer.

The floating gate is isolated from the silicon layer by the oxide layer surrounding it. The electrons are tunneled through this oxide layer, as shown in Fig. 3. Once a charge is added to the floating gate by a programming operation, it is permanently stored there until an erase operation is performed [14], [15]. The effect of these program and erase operations is to change the V_{th} of the cell.

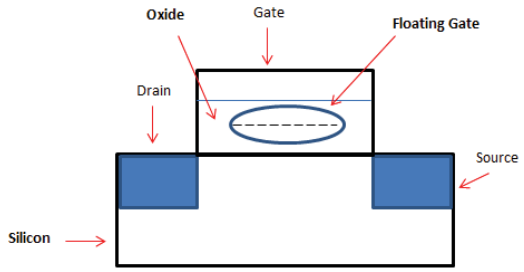

Fig. 3 Floating Gate

NOR is programmed by channel-hot-electron (CHE) injection and erased by Fowler-Nordheim (FN) tunneling [14]. Programming by CHE involves accelerating electrons through the channel between source and drain. These electrons have enough energy to get over the oxide barrier and into the floating gate. Erasing by FN involves applying a high negative voltage to the cell gate with respect to the substrate. This results in the electrons being pulled from the floating gate into the substrate. NAND memory uses FN tunneling for programming and erasing. Programming involves applying a high positive voltage to the cell gate with respect to the substrate. The electrons are then pulled from the substrate into the floating gate.

Within the NAND Flash family, there are three distinct types of memory. SLC (Single Level Cell) can store only 1 bit of data per cell, and can be either programmed (0) or erased (1), as shown in Fig. 4 (a). MLC (Multi-level Cell) stores 2 bits of data per cell in 4 levels - 00 Fully Programmed, 01 Partially Programmed, 10 Partially Erased, 11 Fully Erased, as detailed in Fig. 4 (b).

Finally, TLC (Triple Level Cell) stores 3 bits of data per cell in 8 levels, ranging from 000 Fully Programmed to 111 Fully Erased. The V_{th} distribution arrangements are shown in Fig. 4 (c).

III. Related Research

A. MLC NAND Flash Memory

In order to guarantee reliability for NAND Flash memory, strong ECC is required. But, using stronger ECC leads to a drop in performance, particularly for implementations that sometimes require extra reads for soft-decoding, such as LDPC. Using a simulated SSD which took into account other factors that affect reliability including Bit Error Rate (BER) and the variations between Flash chips, and using endurance/data retention figures found by Cai [16] to calibrate the simulation, a reliability/performance trade-off to estimate the effect of read retry operations on the SSD was investigated [17]. Results showed that the endurance figures given by the manufacturers can be doubled when using read retry. The

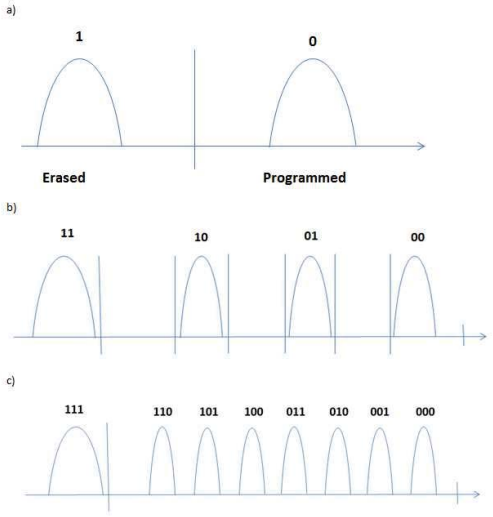

Fig. 4 Voltage Threshold Distribution for SLC, MLC and TLC

retention figures were also shown to be improved. However, with this improvement in the lifetime length of the SSD, comes a drop in performance. Also, when using a single BER all chips in the SSD fail at the same time when they reach the limit of their endurance. A single BER is used to define the read level for one device. But, when using multi BERs, the endurance of the whole SSD is based on the last chip that fails.

Another issue with Flash memory is bad blocks - how to deal with them and how to best use them to further enhance reliability. In Wang and Wong [18], a wear leveling algorithm, Bad Block Salvaging (BBS), reduced the number of worn-out blocks by an average of 46.5%. The current wear leveling methods look at data as either hot or cold, depending on how often it is updated. Hot data is stored in blocks that have not been used very much, while cold data is stored in blocks that have been heavily used. To keep the blocks wearing evenly, blocks with hot data can be swapped for blocks with cold data. However, blocks will still wear out after a certain number of P/E (program/erase) cycles. When the number of these bad blocks reaches the limit defined by the ECC on the chip, the entire chip is considered useless. However, within these bad blocks, there may still be good pages which can be used. BBS divides worn-out blocks into backing blocks, discarded blocks and salavaged blocks, and reuses them as part of wear leveling, with salvaged blocks swopped in to store cold data. The results showed that this technique can lessen worn out blocks by 46.5%.

A Block management scheme, Smart Retirement FTL (SR-FTL), extends Flash memory life by reusing Flash blocks which have been previously cycled to their maximum manufacturer-specified P/E limits. It uses worn out blocks to store any data that has a short retention time, while also managing the reliability of these worn blocks, thereby extending the life of other, unworn blocks [19]. The FTL uses three methods - address mapping, wear leveling, and garbage collection (GC). Currently, the FTL used in Solid State Drives (SSDs) considers any blocks that wear out early as read-only and not to be used further. However, this leaves a gradual shortage of over-provisioning space, as these free blocks are needed to replace the worn-out blocks which increase in number, the more they are used.

When using NAND Flash memory in SSDs, there are a lot of competing goals and many trade-offs and compromises. The minimum number of P/E cycles acceptable for each block on a chip is defined in the JEDEC specification [4], and as such, there already exist certain trade-offs between these defined cycles and the actual time required for data retention - if there is data that has a retention need of less than 1 year, then worn blocks can help by being used to gain extra P/E cycles. ECC can also be tailored for the use case - more powerful or less stringent can be employed for higher or lower endurance requirements respectively. The affect of write amplification on reliability is such that the more space available for over-provisioning, the less GC is called, therefore GC becomes more efficient, and the more often GC is called, the less space is available for over-provisioning, and the more write amplification is affected. Write amplification occurs when GC and wear leveling cause data to be rewritten to the Flash device [20]. Because of how Flash memory works, before writing to a page, the entire block containing that page must first be erased, even if only a single bit is to be changed. Over the lifetime of the device, this additional work affects the reliability of the device [21]. SF-FTL was shown to successfully reduce wear on blocks by 44% to 84% when it is used close to the manufacturer's specified end of life [19].

A different method of extending the life of Flash memory is by using data patterns. DPA (Data Pattern Aware) changes the ratio of the ones and zeros in the data stored on the device, thereby trying to decrease the appearance of patterns that are affected by noise. This, in turn, decreases the overall cell BER, which then increases system endurance [22]. When the interfering cells were programed to 10 or 00, *program* errors happened more often, and when the cells were programed to 00, *retention time* errors happened more often. Two extensions of DPA are:

1) Pattern Probability Unbalance (DPA-PPU) - checks the data ratio between the ones and zeros and depending on how tight this is, uses either "de-correlation", using XOR operations between the neighbouring bytes to reduce the ratio, or "scrambling", to shake up the ratio of ones and zeros
2) Data-Redundancy Management (DPA-DRM) - looks at the redundant bits, as they have more random patterns, and therefore, are more likely to have bit errors. These redundant pages are stored in a block, separate to the data pages, with a mapping table to store the redundant page with its data page. These pages are then improved by using a stronger BCH ECC.

A further extension on DPA-PPU is to change the size of the data chunk, based on the number of P/E cycles - initially, a large data chunk size is used, as the error rate is low, and so the number of scrambling operations is less. This reduces the efficiency of the scrambling. As the error rate grows, however,

the chunk size is reduced, which makes the scrambling more efficient, but also leads to more redundant bits. Experimental results of a 3x or 4x lifetime extensions on MLC Flash memory were shown, with ˜ 13% cost in performance.

Further work using data patterns found a reduction of 36% in BER and 97% interference for 4x-nm Flash memory, and for 2x-nm Flash memory, a reduction of 48% BER and 86% interference [23]. These results were achieved by implementing a data randomisation scheme, based on a pseudorandom generator seeded by address, therefore taking into account all neighbouring cells in every direction. It was found that the interference between cells has much more of an effect on endurance than the effect of P/E cycling, and so the patterns used in testing took into account neighbouring cells on both the horizontal and vertical axis. This scheme did not incur any large performance delays and positively increased endurance in the chips tested.

When attempting to improve reliability of Flash memory, it is also necessary to look at the physical restrictions impeding a tight threshold voltage distribution width in a NAND Flash array. A tight distribution is made possible by using a program-and-verify algorithm, which can move the V_T distributions of a cell to a specific level, by changing the charge in that cell's floating gate. This movement is affected by the program noise (PN) and random telegraph noise (RTN). With the proper use of ECC's, the restriction that affects V_T the most is when charge detraps from the tunnel oxide [24]. Results of testing fresh NAND Flash devices using Incremental Step Pulse Programing (ISPP) to investigate how tight the V_T distributions were during program operations, showed a linear increase between the V_T and the number of pulses applied, because the applied amplitude is always a constant step, V_S. This is another limit on the program accuracy, as became evident by programing cells in a page from the single level (SL) of erased or programed, to the multi level (ML). The results of these program operations showed that PN affects the V_T distribution more with a higher V_S. RTN was almost the same on all of the programed levels. However, the results also showed that PN can be reduced if a smaller V_S is used. It was found that temperature had no affect on either PN or RTN, and that P/E cycles created new oxide traps in the RTN, which then caused even more widening of the V_T distribution.

Results pertaining to the detrapping of charge during idle or bake times show that this also causes a widening of the V_T distribution, due to the stress added by P/E cycles. This was tested by doing two bakes, one after another, and monitoring the V_T distribution of the cells that had previously been programed. The JEDEC specifications were used to simulate real-world use. Charge detrapping causes considerable widening of the erase, L0, distribution and less widening of the three program distributions, L1, L2, and L3. Of these programming distributions, the most widening occurs in the L3 distribution, with some widening in L2 and very little widening in L1.

Other works, such as Mohan et al. [25], have focused more closely on exploring the trade-off between retention and endurance. This was done, firstly, by developing a model incorporating the results of running numerous P/E cycles, along with the affect of recovery on the retention of the memory. Secondly, the model was then used to quantify the trade-off between the retention and endurance of the memory, based on the lifetime use required by large datacentres. Thirdly, a policy was developed to implement the timely refresh of NAND Flash cells in SSDs, thereby further extending the life of the hardware.

Building on previous work related to extending the lifetime of NAND Flash memory by relaxing retention time [25]–[29], WARM (write-hotness aware retention management) policy was devised [30]. This also promotes data refresh, as does [25], by refreshing pages that are most-often written to, i.e. hot pages, and predicts endurance figures when using varying retention times. This policy physically collects together the hot pages in a device, separating them from the cold pages, thereby making less work for the flash controller when choosing hot pages to refresh. When these pages are separated into the hot and cold sets of blocks, each set then has separate policies applied to it.

Extending the lifetime of NAND Flash memory by looking at the possibility of trading retention time for either/both endurance and programing speed is another option [29]. By increasing the accuracy of programing, a decrease in programing speed is caused. This allows for a larger amount of cell noise, leading to longer endurance and, or, data retention. Results show that by reducing retention time to 1 week the endurance increases twice-fold and the normalised programing speed (V_{pp}) increases from 0.2 to 0.345, and by reducing the retention time even further to 1 day, the endurance increases three-fold and the V_{pp} increase to 0.459.

There are also benefits of performing retention relaxation early in the life of the Flash memory, because, as the memory lives on, there is an increase in retention errors [28]. The improvements found could then be used to speed up data writes and use less ECC without a decrease in reliability. An SSD model design was used with options to either improve the speed of write operations or improve the cost of ECC - each returning different retention times. The simulated results showed an increase in write response speed of 1.8-5.7, as well as ECC benefits.

Some work has been carried out using raw flash chips to characterise and analyse NAND Flash memory errors. One such experiment built and used an FPGA-based hardware test platform to perform P/E cycles directly on the chips. These tests involved continually erasing and then programing a block with pseudorandom data, while at room temperature. This resulted in the discovery of distinct errors, which were then characterised and analysed, and found the most prevalent errors were relating to retention [31], [27]. Cai et al. [26] proposed three new methods to extend the time before data becomes too corrupt for ECC to correct it:

1) remap stored data before it has too many errors to correct
2) reprogram data in-place, and then remap if required
3) takes into account the number of P/E cycles a page has already gone through, and changes the rate at which both reprograming and remapping is performed

TABLE I
TLC VALUES

000	Fully Programmed
001	
010	
011	
100	
101	
110	
111	Fully Erased

B. TLC NAND Flash Memory

The theory of a TLC memory cell was proposed in 1997 [32]. This new cell would have a reduced capacity area and efficient ECC. In 1995, a method of increasing the density of the NAND Flash cells was proposed [33], using up to 4-bit cells. This would require narrow V_{th} distributions and high programming speeds.

It is our contention that TLC will suffer from the same problems with reliability as both SLC [2] and MLC [34], but to greater degrees. Instead of having two states, programmed or erased, like in SLC, or four states, like in MLC, there are now eight possible states for TLC, as displayed in Table I, which means there is a far higher chance of V_{th} distributions crossing read boundaries, leading to errors. Because of this, the differences in endurance gradients across blocks and pages in TLC needs to be characterised and quantified.

The layout of a TLC block and the BER on the block and page level in a selection of individual blocks was mapped [35]. This research mapped a TLC page as having a Left and Right MSB page, a Left and Right Central Significant Bit (CSB), and a Left and Right LSB. To do this, firstly a typical layout of a TLC chip was devised. Next, the BER was analysed, both as an average across a number of blocks, and on individual pages in a block. It was discovered that often the state of the cell in question changed from "the highest level to the lowest level", rather than one level at a time. A theory proposed to explain this was that the three bits in a TLC chip were not being programmed at the same time, but instead, one at a time. This meant that if an error occurred in either the first or second bit, the state of the cell would be changed by more than one level. Finally, a new ECC was designed, which would work on all three bits simultaneously.

Reliability is a function of both endurance and retention, and while the work mentioned above focused on ECC design, it tested for endurance only, with no attempt at retention testing. Furthermore, only a sample of blocks was trialled and so, no endurance map applicable across devices could be drawn.

One major issue with all NAND Flash memory is its vulnerability to disturb errors, particularly read-disturbs. These errors appear when one cell is read multiple times, which causes its neighbour cells to lose data. This happens because cells in NAND are connected in a string, and cells in the same string are affected accidentally when changes are made to other cells in the string. With numerous accidental programs to a neighbour cell, the logic state of the cell can be changed. If the bits changed are above the ECC level, then a read-disturb error happens. Read Reclaim (RR) attempts to prevent these read disturbs before they occur [36], and was proven to reduce the overall execution time for read retries by 50%, by finding the blocks that are most often read and moving the data to other, less read, blocks, before a read disturb error occurs. However, this work used an FTL simulator based on trace data, not raw Flash devices.

IV. THE TOOL CHAIN

The tool chain developed for use in this project is comprised of a NAND Flash Utility Tester, Environmental Oven and Graphical User Interface (GUI).

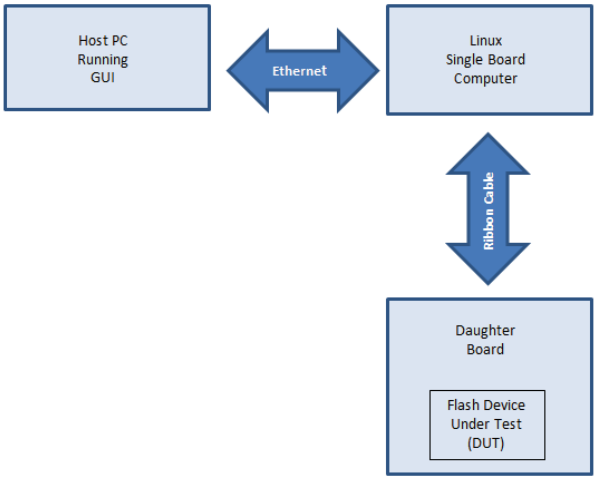

Fig. 5 Overview of test system

As shown in Fig. 5, the GUI is installed on a computer, with the tester units connected via Ethernet cables. These, in turn, are connected to daughter boards, on which the Devices Under Test (DUT) are placed. On initialising the GUI, a TCP/IP connection to the Linux on the tester unit is opened. A grammar of commands are then used in order to run program, read and erase operations, among other operations. The Environmental Oven is used to run temperature controlled test cycles - the oven is ported so the tester units can go directly into these ovens.

V. CURRENT POSITION

To date, the research project has completed a number of phases. Firstly, an Non-Disclosure Agreement (NDA) is in place with a manufacturer. This allowed for the receipt of a batch of preproduction TLC Flash part samples and a preliminary datasheet. Following this, an initial set of tests was performed. These tests allowed us to specify, and write, a new driver requirement for the existing tester. Currently, more specific characterisation tests are underway.

VI. FUTURE WORK

Characterisation of this Flash involves running initial destruct tests, which, as mentioned above, are currently

underway. These tests will cycle blocks until the BER exceeds the ECC limit. This will result in the number of errors found for each test block, and for every page in those blocks, for all 6 page types. They will also give the maximum endurance for each tested block, which occurs when blocks are read immediately after cycling (no retention). Next, the endurance will be found for several retention levels. As the retention time increases, the achievable endurance is expected to decrease. Retention testing is performed by baking the devices for a length of time, calculated using the Arrhenius Equation for Reliability [37]. Following this period the devices will be checked for post-retention errors. This involves comparing the data which was read with the data which was written before the retention bake. Devices from different wafers will be tested, resulting in the endurance versus retention characteristic for this NAND family. This will be repeated with devices from a second NAND vendor.

Further work will be to investigate the impact of data patterns on the reliability of TLC devices, both from the point of view of manipulating the incoming data stream so that high-error inducing data patterns are minimized, and also for developing non-destructive test patterns for accelerated testing.

All of these results will help lay out the optimal trade-off between endurance and retention, depending on the users' desired outcome or application.

VII. Conclusion

The most relevant current literature on NAND Flash memory was presented in this paper, with a focus on NAND Flash memory and reliability, along with Flash memory life extension - including read-disturb management [36], bad blocks management [18] and [19], the use of data patterns [22] and [23], physical issues when fitting cells into narrow V_T intervals [24], reliability/performance trade-offs [17], [28]-[30], and ECCs [35]. Of this work, only two worked with raw Flash chips [31], [27], albeit MLC (Multi-Level Cell) NAND Flash memory. Most of this current work reviewed used various Flash Translation Layer (FTL) and SSD (Solid State Drive) simulators and models.

The work reviewed is not all the current research on NAND Flash memory using TLC, nor is it all the research on the reliability of NAND Flash memory, but it is all that is relevant to this project. TLC Flash memory was used by only two research groups - one looked at ECC [35], and the other at read disturb errors [36], with the all testing carried out using a simulator. Very little relevant research used raw Flash chips, and at the time of writing, no relevant current published research has been conducted on TLC raw chips. These papers show that there is still a large gap in the knowledge relating to TLC, both using raw flash chips, and reliability trade-off incorporating both TLC and raw chips. Furthermore, with the imminent introduction of 3D geometry by many manufacturers, and its associated increase in nano-scale structures such as floating gates, TLC cells are currently set to be the cell architecture of choice going forward [38].

Acknowledgment

The authors would like to thank this paper reviewers, along with Barry Fitzgerald.

References

[1] P. Pavan, R. Bez, P. Olivo, and E. Zanoni, "Flash Memory Cells - An Overview," *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1248 –1271, aug 1997. (Online). Available: http://0-ieeexplore.ieee.org.mislibsrv.lit.ie/stamp/stamp.jsp?tp=&arnumber=622505

[2] S. Aritome, R. Shirota, G. Hemink, T. Endoh, and F. Masuoka, "Reliability issues of flash memory cells," *Proceedings of the IEEE*, vol. 81, no. 5, pp. 776 –788, may 1993. (Online). Available: http://0-ieeexplore.ieee.org.mislibsrv.lit.ie/stamp/stamp.jsp?tp=&arnumber=220908

[3] IEEE, "IEEE Standard Definitions and Characterization of Floating Gate Semiconductor Arrays," *IEEE Std 1005-1998*, 1998, endurance: Pg 86, Section 7.

[4] JEDEC, *Stress-Test-Driven Qualification of Integrated Circuits - JESD47H-01*, Jedec Solid State Technology Association, Published by JEDEC Solid State Technology Association 2011 3103 North 10th Street, Suite 240 South Arlington, VA 22201, APRIL 2011.

[5] C. Compagnoni, C. Miccoli, R. Mottadelli, S. Beltrami, M. Ghidotti, A. Lacaita, A. Spinelli, and A. Visconti, "Investigation of the Threshold Voltage Instability after Distributed Cycling in Nanoscale NAND Flash Memory Arrays," in *Reliability Physics Symposium (IRPS), 2010 IEEE International*, may 2010, pp. 604 –610. (Online). Available: http://0-ieeexplore.ieee.org.mislibsrv.lit.ie/stamp/stamp.jsp?tp=&arnumber=5488762

[6] R. Micheloni, A. Marelli, and R. Ravasio, *Error Correction Codes for Non-Volatile Memories*. Springer, 1998, vol. XII. (Online). Available: http://www.springer.com/engineering/circuits+%26+systems/book/978-1-4020-8390-7

[7] G. Dong, N. Xie, and T. Zhang, "On the Use of Soft-Decision Error-Correction Codes in NAND Flash Memory," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 58, no. 2, pp. 429–439, Feb 2011. (Online). Available: http://0-ieeexplore.ieee.org.mislibsrv.lit.ie/stamp/stamp.jsp?tp=&arnumber=5629456

[8] E. Yeo, "AN LDPC-Enabled Flash Controller in 40nm CMOS," Santa Clara, CA: Flash Memory Summit, 2012. (Online). Available: http://www.flashmemorysummit.com/English/Collaterals/Proceedings/2012/20120822_TE22_Yeo.pdf

[9] J. Wang, T. Courtade, H. Shankar, and R. Wesel, "Soft Information for LDPC Decoding in Flash: Mutual-Information Optimized Quantization," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, Dec 2011, pp. 1–6. (Online). Available: http://0-ieeexplore.ieee.org.mislibsrv.lit.ie/stamp/stamp.jsp?tp=&arnumber=6134417

[10] S. K. Tewksbury and J. E. Brewer, *Nonvolatile Memory Technologies with Emphasis on Flash*, ser. IEEE Press Series on Microelectronic Systems, J. E. Brewer and M. Gill, Eds. 445 Hoes Lane, Piscataway, NJ 08854: IEEE Press Series, 2008.

[11] P. Desnoyers, "Empirical Evaluation of NAND Flash Memory Performance," *SIGOPS Oper. Syst. Rev.*, vol. 44, no. 1, pp. 50–54, Mar. 2010. (Online). Available: http://doi.acm.org/10.1145/1740390.1740402

[12] Statista. (2015) Global market share held by NAND Flash memory manufacturers worldwide from 1st quarter 2010 to 3rd quarter 2015. The Statistics Portal. (Online). Available:http://www.statista.com/statistics/275886/market-share-held-by-leading-nand-flash-memory-manufacturers-worldwide

[13] J. H. Yoon, "3D NAND Technology Implications to Enterprise Storage Applications," Santa Clara, CA, USA: Flash Memory Summit, August 2015. (Online). Available: http://www.flashmemorysummit.com/English/Collaterals/Proceedings/2015/20150811_FM12_Yoon.pdf

[14] R. Bez, E. Camerlenghi, A. Modelli, and A. Visconti, "Introduction to Flash Memory," *Proceedings of the IEEE*, vol. 91, no. 4, pp. 489 – 502, april 2003. (Online). Available: http://0-ieeexplore.ieee.org.mislibsrv.lit.ie/stamp/stamp.jsp?tp=&arnumber=1199079

[15] P. Hasler and T. Lande, "Overview of floating-gate devices, circuits, and systems," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 48, no. 1, pp. 1 –3, jan 2001. (Online). Available: http://0-ieeexplore.ieee.org.mislibsrv.lit.ie/stamp/stamp.jsp?tp=&arnumber=913180

[16] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Threshold Voltage Distribution in MLC NAND Flash Memory: Characterization, Analysis and Modeling," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, 2013, pp. 1285–1290. (Online). Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6513712&queryText%3Dthreshold+voltage+distribution+in+mlc+nand+flash+memory%3A+characterization%2C+analysis+and+modeling

[17] L. Zuolo, C. Zambelli, R. Micheloni, D. Bertozzi, and P. Olivo, "Analysis of reliability/performance trade-off in Solid State Drives ," in *Reliability Physics Symposium, 2014 IEEE International*, June 2014, pp. 4B.3.1–4B.3.5.

[18] C. Wang and W.-F. Wong, "Extending the Lifetime of NAND Flash Memory by Salvaging Bad Blocks," in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '12. San Jose, CA, USA: EDA Consortium, 2012, pp. 260–263. (Online). Available: http://dl.acm.org/citation.cfm?id=2492708.2492773

[19] P. Huang, G. Wu, X. He, and W. Xiao, "An Aggressive Worn-out Flash Block Management Scheme to Alleviate SSD Performance Degradation," in *Proceedings of the Ninth European Conference on Computer Systems*, ser. EuroSys '14. New York, NY, USA: ACM, 2014, pp. 22:1–22:14. (Online). Available: http://doi.acm.org/10.1145/2592798.2592818

[20] X.-Y. Hu, E. Eleftheriou, R. Haas, I. Iliadis, and R. Pletka, "Write Amplification Analysis in Flash-based Solid State Drives," in *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference*, ser. SYSTOR '09. New York, NY, USA: ACM, 2009, pp. 10:1–10:9. (Online). Available: http://doi.acm.org/10.1145/1534530.1534544

[21] R. Agarwal and M. Marrow, "A closed-form expression for write amplification in NAND Flash," in *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, Dec 2010, pp. 1846–1850. (Online). Available: http://0-ieeexplore.ieee.org.mislibsrv.lit.ie/stamp/stamp.jsp?tp=&arnumber=5700261

[22] J. Guo, Z. Chen, D. Wang, Z. Shao, and Y. Chen, "DPA: A data pattern aware error prevention technique for NAND flash lifetime extension," in *Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific*, Jan 2014, pp. 592–597. (Online). Available: http://0-ieeexplore.ieee.org.mislibsrv.lit.ie/xpl/articleDetails.jsp?arnumber=6742955&queryText=DPA%3A+A+data+pattern+aware+error+prevention+technique+for+NAND+flash+lifetime+extension&newsearch=true&searchField=Search_All

[23] J. Cha and S. Kang, "Data Randomization Scheme for Endurance Enhancement and Interference Mitigation of Multilevel Flash Memory Devices," *ETRI Journal*, vol. 35, no. 1, pp. 166–169, February 2013. (Online). Available: http://dx.doi.org/10.4218/etrij.13.0212.0273http://etrij.etri.re.kr/etrij/journal/article/article.do?volume=35&issue=1&page=166

[24] G. Paolucci, C. Compagnoni, A. Spinelli, A. Lacaita, and A. Goda, "Fitting Cells Into a Narrow V_T Interval: Physical Constraints Along the Lifetime of an Extremely Scaled NAND Flash Memory Array," *Electron Devices, IEEE Transactions on*, vol. 62, no. 5, pp. 1491–1497, May 2015. (Online). Available: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=7089352&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D7089352

[25] V. Mohan, S. Sankar, and S. Gurumurthi, "reFresh SSDs: Enabling High Endurance, Low Cost Flash in Datacenters," University of Virginia, Department of Computer Science University of Virginia Charlottesville, VA 22904, Tech. Rep., May 2012. (Online). Available: http://www.cs.virginia.edu/~vm9u/files/RefreshSSDs.pdf

[26] Y. Cai, G. Yalcin, O. Mutlu, E. Haratsch, A. Cristal, O. Unsal, and K. Mai, "Flash correct-and-refresh: Retention-aware error management for increased flash memory lifetime," in *Computer Design (ICCD), 2012 IEEE 30th International Conference on*, 2012, pp. 94–101. (Online). Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6378623&queryText%3Dflash+correct-and-refresh%3A+retention-aware+error+management+for+increased+flash+memory+lifetime

[27] Y. Cai, G. Yalcin, O. Mutlu, E. Haratsch, A. Cristal, O. Unsal, and K. Mai, "Error Analysis and Retention-Aware Error Management for NAND Flash Memory," Intel, Technology Journal Paper 8, Volume 17, Issue 1, 2013. (Online). Available:https://noggin.intel.com/system/files/article/paper-8-error-analysis-and-retention-aware-error-management-for-nand-flash-memory.pdf

[28] C.-L. Y. Ren-Shuo Liu and W. Wu, "Optimizing NAND Flash-Based SSDs via Retention Relaxation," 10th USENIX Conference on File and Storage Technologies. San Jose, CA: USENIX FAST, February 15th 2012. (Online). Available: http://static.usenix.org/events/fast/tech/full_papers/Liu.pdf

[29] Y. Pan, G. Dong, Q. Wu, and T. Zhang, "Quasi-nonvolatile SSD: Trading flash memory nonvolatility to improve storage system performance for enterprise applications," in *High Performance Computer Architecture (HPCA), 2012 IEEE 18th International Symposium on*, Feb 2012, pp. 1–10. (Online). Available: http://0-ieeexplore.ieee.org.mislibsrv.lit.ie/xpl/articleDetails.jsp?arnumber=6168954&queryText=Quasi-nonvolatile+ssd%3A+Trading+flash+memory+nonvolatility+to+improve+storage+system+performance+for+enterprise+applications&newsearch=true&searchField=Search_All

[30] Y. Luo, Y. Cai, S. Ghose, J. Choi, O. Mutlu, "WARM: Improving NAND flash memory lifetime with write-hotness aware retention management," in *(2015, May). . In Mass Storage Systems and Technologies (MSST), 2015 31st Symposium on (pp. 1-14). IEEE.*, 2015. (Online). Available: https://users.ece.cmu.edu/~omutlu/pub/warm-flash-write-hotness-aware-retention-management_msst15.pdf

[31] Y. Cai, E. Haratsch, O. Mutlu, and K. Mai, "Error patterns in MLC NAND flash memory: Measurement, characterization, and analysis," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2012*, march 2012, pp. 521 –526. (Online). Available: http://0-ieeexplore.ieee.org.mislibsrv.lit.ie/stamp/stamp.jsp?tp=&arnumber=6176524

[32] T. Tanaka, T. Tanzawa, and K. Takeuchi, "A 3.4-Mbyte/sec Programming 3-Level NAND Flash Memory Saving 40% Die Size Per Bit," Symposium on VLSI Circuits Digest of Technical Papers, Tech. Rep. 4-93081 3-76-X, 14-14 June 1997, pages 65 - 66. (Online). Available: http://www.lsi.t.u-tokyo.ac.jp/image/publicationIC7.pdf

[33] G. Hemink, T. Tanaka, T. Endoh, S. Aritome, and R. Shirota, "Fast and accurate programming method for multi-level NAND EEPROMs," in *VLSI Technology, 1995. Digest of Technical Papers. 1995 Symposium on*, jun 1995, pp. 129 –130. (Online). Available: http://0-ieeexplore.ieee.org.mislibsrv.lit.ie/stamp/stamp.jsp?tp=&arnumber=520891

[34] L. Grupp, A. Caulfield, J. Coburn, S. Swanson, E. Yaakobi, P. Siegel, and J. Wolf, "Characterizing Flash Memory: Anomalies, Observations, and Applications," in *Microarchitecture, 2009. MICRO-42. 42nd Annual IEEE/ACM International Symposium on*, dec. 2009, pp. 24 –33. (Online). Available: http://0-ieeexplore.ieee.org.mislibsrv.lit.ie/stamp/stamp.jsp?tp=&arnumber=5375312

[35] E. Yaakobi, L. Grupp, P. Siegel, S. Swanson, and J. Wolf, "Characterization and Error-Correcting Codes for TLC Flash Memories," in *Computing, Networking and Communications (ICNC), 2012 International Conference on*, Jan 2012, pp. 486–491. (Online). Available: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6167470&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6167470

[36] K. Ha, J. Jeong, and J. Kim, "A Read-disturb Management Technique for High-density NAND Flash Memory," in *Proceedings of the 4th Asia-Pacific Workshop on Systems*, ser. APSys '13. New York, NY, USA: ACM, 2013, pp. 13:1–13:6. (Online). Available: http://doi.acm.org/10.1145/2500727.2500743

[37] JEDEC ASSOCIATION, *Method for Developing Acceleration Models for Electronic Component Failure Mechanisms - JESD91A*, JESD91A - (Revision of JESD91) ed., JEDEC, AUGUST 2003, page 8 - Arrhenius equation definition. (Online). Available: http://www.jedec.org/sites/default/files/docs/jesd91a.pdfhttp://www.jedec.org/standards-documents/dictionary/terms/arrhenius-equation-reliability

[38] J. Choe, "Comparison of 20nm & 10nm-class 2D Planar NAND and 3D V-NAND Architecture." Santa Clara, CA, USA: Fash Memory Summit, August 2015. (Online). Available: http://www.flashmemorysummit.com/English/Collaterals/Proceedings/2015/20150811_FM12_Choe.pdf

Sorcha Bennett has a BSc. in Computer Systems, a HDip. in Middleware Software Development, both from University of Limerick, Limerick, Ireland, and is currently working towards a Ph.D. in the Department of Electrical and Electronic Engineering, Limerick Institute of Technology, Limerick, Ireland. Her research interests include low-level communication with Flash devices, test design and pattern usage.

Joe Sullivan has a BEng. from University of Brighton, England, and a Ph.D. degree in Computer Science from University of Limerick, Limerick, Ireland. He has been a Senior Lecturer in Limerick Institute of Technology, Limerick, Ireland, for the last 14 years. His background is in engineering and he spent more than ten years with Analog Devices, specializing in Non Volatile Memory. His current interests are low-level interaction with SSDs, including socket and interface design, and machine learning.