SECURITY AND PRIVACY THREAT OF SCRUBBED NAND FLASH MEMORY AND COUNTERMEASURE

by

MD MEHEDI HASAN

A THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering in The Department of Electrical and Computer Engineering to The School of Graduate Studies of The University of Alabama in Huntsville

HUNTSVILLE, ALABAMA

2020
In presenting this thesis in partial fulfillment of the requirements for a master's degree from The University of Alabama in Huntsville, I agree that the Library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by my advisor or, in his absence, by the Chair of the Department or the Dean of the School of Graduate Studies. It is also understood that due recognition shall be given to me and to The University of Alabama in Huntsville in any scholarly use which may be made of any material in this thesis.

[Signature]

MD MEHEDI HASAN  
04/01/2020 (date)
THESIS APPROVAL FORM

Submitted by MD MEHEDI HASAN in partial fulfillment of the requirements for the degree of Master of Science in Engineering in Computer Engineering and accepted on behalf of the Faculty of the School of Graduate Studies by the thesis committee.

We, the undersigned members of the Graduate Faculty of the University of Alabama in Huntsville, certify that we have advised and/or supervised the candidate of the work described in this thesis. We further certify that we have reviewed the thesis manuscript and approve it in partial fulfillment of the requirements for the degree of Master of Science in Engineering in Computer Engineering.

04/02/2020
Dr. Biswajit Ray
(Date)
Committee Chair

April 2, 2020
Dr. Timothy Boykin
(Date)
Committee member

April 3, 2020
Dr. David Pan
(Date)
Committee member

Ravi Gorur
Digitally signed by Ravi Gorur
Date: 2020.04.06 17:10:07 -05'00'

Dr. Ravi Gorur
(Date)
Department Chair

Shankar Mahalingam
Digitally signed by Shankar Mahalingam
DN: cn=Shankar Mahalingam, ou=College of Engineering, ou=Dean, email=sm0026@uah.edu, c=US
Date: 2020.04.08 10:58:05 -05'00'

Dr. Shankar Mahalingam
(Date)
College Dean

David Berkowitz
Digitally signed by David Berkowitz
DN: cn=David Berkowitz, c=University of Alabama in Huntsville, ou=Graduate Dean, email=d berkowitz@uah.edu, c=US
Date: 2020.04.08 14:36:29 -05'00'

Dr. David Berkowitz
(Date)
Graduate Dean
ABSTRACT

The School of Graduate Studies
The University of Alabama in Huntsville

Degree Master of Science in Engineering
College/Dept. Engineering/Electrical and Computer Engineering
Name of Candidate Md Mehedi Hasan
Title Security and privacy threat of scrubbed NAND flash memory and countermeasure

Due to the high bit density and low-cost, NAND flash memory has grown popularity of various consumer electronics over the past decade. NAND flash memory is used as the primary storage medium of smartphones, USB drives, SSD and laptops. To ensure the high bit density and low cost, there is a tradeoff with security and reliability. To expedite the performance of a NAND flash and to mitigate the early wear-out, erase is not performed frequently. As erase is a costly operation that reduces the lifetime of a flash memory cell, a page-level digital sanitization is used to erase data temporarily from a NAND block, which is known as scrubbing. But as a matter of concern, this page level scrubbing based sanitization generates security and privacy issues because of the cell threshold variation due to date retention effect. An adversary can utilize the data retention property to recover the sanitized data. In this thesis, we demonstrate a data recovery procedure from the commercial flash memory chip, sanitized with scrubbing, using partial erase operation. Our result demands the necessity of analog scrubbing and we propose and implement a partial program based analog scrubbing method to make scrubbing based delete data unrecoverable.

NAND flash has the full potential to be used as a computing device. For example, NAND flash can be primarily used as a weight storage device for edge computing where
power is a big constraint. To use flash memory as a weight storage device, lots of reliability issues need to be addressed. Over time flash memory is changing from its 2D planar structure to a vertically stacked structure. This high-density memory is increasing its bit density with the cost of reliability. We explore the other two major opportunities of NAND flash to enhance the computing reliability issues and big block management issues for high density stacked NAND flash. To exploit these opportunities in improving flash memory reliability in computing and lifetime, we also perform various experiments using the flash memory device. In this thesis, we further (1) explore the reliability issues while NAND flash memory is used as a weight storage device for edge computing and propose a page priority based weight storage method so that the accuracy can be maintained at an acceptable level, and (2) observe the layer-based dependency of BER on high-density NAND flash due to the incoherent erase operation and suggest a new method for big block management to expedite the lifetime of a flash memory.

Overall, this thesis depends on the understanding of the basic operations and security vulnerabilities of NAND flash memory through rigorous experimental characterization and proposes new methods to improve the security hole along with reliability issues.

Abstract Approval:  Committee Chair
Dr. Biswajit Ray

Department Chair
Ravi Gorur

Graduate Dean
Dr. Ravi Gorur

David Berkowitz

Dr. David Berkowitz
ACKNOWLEDGMENTS

First and foremost, I am thankful to Allah, the most gracious, most merciful for all he has gifted me with. It is His plan and my belief in Him that helped me finish this work.

I would like to express my gratitude to my advisor Dr. Biswajit Ray for the continuous support and intellectual guidance, motivations, for his patience and immense knowledge and always trusting me to do good work.

I am grateful to my thesis committee members: Dr. Timothy Boykin and Dr. David Pan for their valuable time, cooperation and suggestions.

I would also like to thank my lab mates and Zabed Iqbal who supported directly or indirectly throughout my graduation at the University of Alabama in Huntsville.

I would like to acknowledge the enormous love and support that I received from my family and friends. Firstly, I would like to thank my wife, Jarin Firose Moon, for her enormous encouragement and support in every possible way to do the right thing. Then I would like to thank my friends: Nayeem, Sabbir, Pritom, Sumon, Abir, Nayan, and their family, and many others who really made me happy during my hardest time. I thank my brother and sister and their family for their love and support. Finally, I express my profound gratitude to my parents: Md Badsha Mia and Mst Sazeda Begum, who raised me to be the person I am today.
This thesis paper is dedicated to Bangabandhu Sheikh Mujibur Rahman - father of the nation of Bangladesh
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>xv</td>
</tr>
<tr>
<td>LIST OF SYMBOLS AND ACRONYMS</td>
<td>xvi</td>
</tr>
<tr>
<td>Chapter 1: Introduction</td>
<td>17</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>17</td>
</tr>
<tr>
<td>1.2 Thesis organization</td>
<td>17</td>
</tr>
<tr>
<td>Chapter 2: Background of NAND Flash Memory</td>
<td>19</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>19</td>
</tr>
<tr>
<td>2.2 NAND flash memory</td>
<td>19</td>
</tr>
<tr>
<td>2.2.1 Data storing process in a flash memory cell:</td>
<td>19</td>
</tr>
<tr>
<td>2.2.2 NAND flash array:</td>
<td>21</td>
</tr>
<tr>
<td>2.2.3 Read, program and erase operations:</td>
<td>23</td>
</tr>
<tr>
<td>2.2.4 3D NAND flash memory:</td>
<td>25</td>
</tr>
<tr>
<td>Chapter 3: Security and Privacy of Scrubbed NAND Flash Memory</td>
<td>26</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>26</td>
</tr>
<tr>
<td>3.2 Background</td>
<td>29</td>
</tr>
<tr>
<td>3.2.1 Flash memory organization and operations:</td>
<td>29</td>
</tr>
<tr>
<td>3.2.2 Flash translation layer</td>
<td>31</td>
</tr>
<tr>
<td>3.3 Threat model and assumptions</td>
<td>32</td>
</tr>
<tr>
<td>3.4 Data retrieval after scrubbing</td>
<td>33</td>
</tr>
<tr>
<td>3.4.1 Data retention (DR):</td>
<td>33</td>
</tr>
<tr>
<td>3.4.2 Scrubbing after data retention:</td>
<td>34</td>
</tr>
<tr>
<td>3.4.3 Partial erase:</td>
<td>34</td>
</tr>
<tr>
<td>3.4.4 Data recovery with partial erase:</td>
<td>35</td>
</tr>
<tr>
<td>3.5 Analog scrubbing</td>
<td>38</td>
</tr>
<tr>
<td>3.5.1 Partial page program:</td>
<td>39</td>
</tr>
<tr>
<td>3.5.2 Analog scrubbing with partial page program:</td>
<td>39</td>
</tr>
<tr>
<td>3.6 Implementation and evaluation</td>
<td>41</td>
</tr>
<tr>
<td>3.6.1 Experimental set-up:</td>
<td>41</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figures</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1: Usage of NAND flash memory</td>
<td>18</td>
</tr>
<tr>
<td>Figure 2.1: Cross-sectional view of a floating-gate MOSFET</td>
<td>20</td>
</tr>
<tr>
<td>Figure 2.2: Histogram representation of cell threshold voltage of (a) SLC and (b) MLC NAND flash memory</td>
<td>21</td>
</tr>
<tr>
<td>Figure 2.3: The hierarchical storage in NAND flash array consisting of kilo-bytes of memory cells and the WL electrically connects those cells (called a page of information). Each block consists of multiple WLs. The select gate transistors can be standard MOSFET or FG transistors, depending on manufacturer or technology node</td>
<td>22</td>
</tr>
<tr>
<td>Figure 2.4: NAND flash basic operations. Applied voltage to bitline to (a) read, (b) program, and (c) erase</td>
<td>23</td>
</tr>
<tr>
<td>Figure 2.5: (a) Planner flash memory cell vs (b) 3-D cell geometry which is a gate-all-around vertical channel MOSFET</td>
<td>25</td>
</tr>
<tr>
<td>Figure 3.1: (a) A floating gate (FG) NAND flash memory cell which stores information in the form of charge on the FG. Metal word-line (WL) act as the control gate of the FG transistor. Charge is injected on the FG through tunneling of electron from Si-channel to FG. Blocking oxide prevents back tunneling of electron to control gate. (b) Typical histogram representation threshold voltage for erase state cell and programmed state cell.</td>
<td>30</td>
</tr>
<tr>
<td>Figure 3.2: (a) The architecture of a flash-based storage system. (b) File system overview of storage system</td>
<td>32</td>
</tr>
</tbody>
</table>
Figure 3.3: (a) Histogram representation of threshold voltage of memory cells for stored ‘1’ and ‘0’. Programmed state has a higher threshold than the erased state. (b) Down-shift of threshold histogram of programmed state due to data retention effect. (c) Threshold voltage histogram difference between weak ‘0’ and strong ‘0’. The newly programmed cell has a higher threshold than the old one.

Figure 3.4: Data recovery from scrubbed pages. (a) Original Einstein image (460 × 600) which is stored to the NAND flash. (b) Original image after data retention takes place (c) Scrubbed image, this is all ‘0’ image. We bake the chip for 3 hours in an oven of temperature 120°C in order to accelerate the data retention effect. (d) The raw image recovered using partial erase operation. (e), (f), (g) & (h) Histogram representation of threshold voltage for corresponding image of (a), (b), (c) & (d).

Figure 3.5: (a) Illustration of bit accuracy of a recovered image vs partial erase time. The image was scrubbed with zero-overwrite method after 3 hours of bake at 120°C. (b) For clear illustration, we plot the percentage of weak zeros that flipped into ones as well as the percentage of strong zeros that remained at zero state with respect to erase time.

Figure 3.6: Bit recovery accuracy versus bake time for stored Einstein image on two different NAND flash chips.

Figure 3.7: Analog scrubbing output (a) Stored original Einstein image (460 × 600). (b) Scrubbed image after data retention takes place. We bake the chip for 3 hours in an oven of temperature 120°C in order to accelerate the data retention effect. (c) Recovered raw image using analog scrubbing operation.

Figure 4.1: Neural Network (NN) properties: (a) Fully connected layer in a neural network, where an input layer is connected to a hidden layer and hidden layer is connected to the
output layer. One node is connected to other nodes via weight \((W_{ij})\). Each weight is represented as an 8-bit fixed point data format, (b) Although normalized NN weights range from -1 to 1 but most of the NN weights are near 0, (c) Sample input digits and sample output digits after testing. The check marked digits are classified perfectly and the cross marked digit is classified wrongly. We calculate the accuracy by counting the wrongly classified images from MNIST data-set.

Figure 4.2: Experimental setup to interface NAND flash memory with PC.

Figure 4.3: Reliability issues on stored NN weights in NAND Flash memory based on usage: (a) Measured BER with respect to bake time increases with higher program/erase cycle. This figure illuminates that although higher PE has a significant effect on BER, a 2% usage (max endurance ~ 5000 PE) is quite acceptable in terms of NN weights’ accuracy, (b) Recognition accuracy for MNIST dataset for different memory usage. We find till 100 PE cycle NN accuracy remains acceptable with bake time. However, for PE > 500, data retention effect becomes significant, which conspicuously reduces the detection accuracy of MNIST dataset.

Figure 4.4: BER and resulting recognition accuracy as a function of TID for 2-D (a) and 3-D (b) NAND flash.

Figure 4.5: Classification accuracy: (a) Experimental recognition accuracy versus BER while traditional storage method is used to store the NN weights in flash memory (b) Theoretical average recognition accuracy versus BER while error added randomly to neural network weight for MNIST dataset where each data point shows minimum and maximum value for that specific BER.
Figure 4.6: Neural Network (NN) performance analysis based on BER and flash page to page BER variation: (a) Recognition accuracy get significantly reduced when random error is added to the 1st and 2nd most significant bits whereas recognition accuracy is resilient to least significant bit errors, (b) NAND flash page to page BER variation, LSB page BER is higher than the MSB page BER. ................................................................. 60

Figure 4.7: Weight storage method: (a) Traditional weight storage method where weight bits are stored sequentially in flash memory page, (b) Proposed weight storage method where critical weight bits are stored to the low BER page first and less critical weight bits are stored to the high BER page. ................................................................. 61

Figure 4.8: Comparative recognition accuracy: proposed method shows higher recognition accuracy than traditional storage method. ................................................................. 63

Figure 5.1: Number of pages per block vs technology node. With time number of pages in block increases significantly, (b) 3D NAND flash endurance vs layer. The top and bottom layers have less endurance than the middle layers. ................................................................. 67

Figure 5.2: (a) 3D NAND flash stack based layered structure where the bottom layers have more diameter than the top layers which makes it tapered. (b) The architecture and file system overview of a flash-based storage system................................................................. 69

Figure 5.3: PE cycling and data retention effect on 3D NAND flash in terms of BER vs layer number: (a) For a fresh block without any data retention effect, BER is very low and it does not show layer dependency (b) A 1000 PE cycled block shows layers dependent BER. (c) Fresh block with 150°C HTDR shows little layer dependency on BER. (d) A 1000 PE cycled block with 150°C HTDR shows huge variation of BER in different layers. ........................................................................................................ 72
Figure 5.4: Erase speed variation analysis of 3D NAND flash: (a) Middle layers are slow to erase than the top and bottom layers. This makes the top and bottom layers cell deeply erased (b) Erase operation from bottom to top layers in a 3D NAND layered structure. (c) Non-erased bits vs PE cycle count shows with the increase of PE cycle count, erase becomes slower. (d) Uneven cell threshold distribution of different layers cell in 3D NAND flash.

Figure 5.5: Proposed layer to layer wear leveling technique. Up to a certain PE cycle, every page can be used but after that from the top and bottom layers gradually the most vulnerable pages (MSB pages) should be discarded to retain the block from being weard-out.

Figure 5.6: Layer based BER vs PE cycle count: When cycling incorporated with HTDR effect then top and bottom layers show significantly higher BER than the middle layers.

Figure 5.7: BER using shifted read of a segment (a) second lowest BER using shifted read which confirms permanent trend of layer dependency, (b) lowest BER shifted read option also shows the layer dependency of BER.
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Tables</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 3.1: Evaluation of bit recovery efficiency on different NAND chips</td>
<td>41</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ACRONYMS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FG-MOSFET</td>
<td>Floating gate-MOSFET</td>
</tr>
<tr>
<td>$V_t$</td>
<td>Threshold voltage</td>
</tr>
<tr>
<td>SLC</td>
<td>Single level cell</td>
</tr>
<tr>
<td>MLC</td>
<td>Multi-level cell</td>
</tr>
<tr>
<td>TLC</td>
<td>Triple level cell</td>
</tr>
<tr>
<td>QLC</td>
<td>Quadruple level cell</td>
</tr>
<tr>
<td>SSL</td>
<td>String select line</td>
</tr>
<tr>
<td>GSL</td>
<td>Ground select line</td>
</tr>
<tr>
<td>BL</td>
<td>Bitline</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial neural network</td>
</tr>
<tr>
<td>FTL</td>
<td>Flash translation layer</td>
</tr>
<tr>
<td>BER</td>
<td>Bit error rate</td>
</tr>
</tbody>
</table>
Chapter 1: Introduction

1.1 Motivation

NAND flash memories are becoming increasingly important and used as primary persistent storage devices due to their higher bit density and low cost. The more the data generated the more flash is needed. As a matter of concern, this huge amount of data requires efficient processing and needs to be kept secure. Thus, maintaining data privacy is very much important while there is a chance for the data breach. However, the conventional flash memory has several security issues as efficient performance requires some compromised operation. One of the operations in NAND flash memory is scrubbing based erase operation. In this thesis, we particularly focused on this operation and find security vulnerability and its countermeasure. Also, NAND flash is used in different sensor nodes and edge devices (e.g., cameras, drones, gyroscopes) requiring in-place processing of large input data (see Figure 1.1). We also focused on the reliability issues of NAND flash memory on computing. One major constraint for today’s neural network accelerators is the storage of a large number of model weights (> megabytes of storage space). Flash memory is an attractive choice for storing a large number of ANN weights in edge devices as it is low cost and has a very high bit density. Also, this highly dense flash memory requires efficient block management to enhance the performance and maximize its usage.

1.2 Thesis organization

This thesis is composed of 6 chapters including this introductory chapter. Chapter 2 describes the background of NAND flash memory where basic flash memory structure
and operations are discussed. Chapter 3 explores the security vulnerabilities and countermeasures. Chapter 4 discussed the computing opportunities of NAND flash memory while being used as a weight storage device and reliability issues on NAND flash for computing. Chapter 5 explores the efficient management of big blocks of a high-density NAND flash memory. Finally, chapter 6 draws a summary of this work and suggests future research directions.
Chapter 2: Background of NAND Flash Memory

2.1 Introduction

Flash memory is a non-volatile memory storage device that can be electrically erased and reprogrammed. Based on NAND and NOR logic gates the two main types of flash memory are named. A single flash memory cell, consisting of floating-gate MOSFET (floating-gate metal-oxide-semiconductor field-effect transistor), exhibit internal characteristics similar to those of the corresponding gates. The NAND type is found primarily in USB flash drives, memory cards, and solid-state drives, for general storage and transfer of data.

2.2 NAND flash memory

NAND flash memory is very popular due to its low cost and high bit density. There are lots of reason that affects NAND flash memory security and make this device vulnerable to intruders. Also, for a computing device, there are a lot of reliability issues for which performance becomes degraded. In this section, we present a brief discussion on these topics along with NAND flash memory data storage properties to make the reader more comfortable about our key contributions to this work. We first discuss the process of data storing in NAND flash and introduce the read, write and erase operation of it. Then we introduce the reliability issues of flash memory.

2.2.1 Data storing process in a flash memory cell:
Data stored in a flash memory cell in terms of the threshold voltage of floating gate MOSFET. Floating gate MOSFET is also known as a floating gate transistor. It is a type of MOSFET where the gate is electrically isolated. The floating gate is completely surrounded by highly resistive material, thus the charge contained in it remains unchanged for long periods of time. Generally, Fowler-Nordheim tunneling and hot-carrier injection mechanisms are used to modify the charge stored in the floating gate. Figure 2.1 shows a cross-section of a floating gate (FG) transistor. The gate that lies on the top of a flash cell is known as a control gate (CG) and the gate below of this is the floating gate. On the top of the FG, there is an inter-poly oxide layer and at the bottom of the FG there is a tunnel oxide layer, that’s how this gate becomes isolated from CG and channel or substrate. As a result, when the electrons become trapped into the FG, they remain there in the FG even when the flash memory is powered off, thus making it a non-volatile memory cell.

Flash memory has gone through a transition from single-level cell (SLC) to multi-level cell (MLC) to triple-level cell (TLC) to quadruple level cell (QLC) meaning that from

![Figure 2.1 Cross-sectional view of a floating-gate MOSFET](image-url)
one cell we can extract one bit for SLC, 2 bit for MLC, 3 bit for TLC, and 4 bit for QLC. Which on the other hand increases the density at the cost of a cell threshold voltage. Figure 2.2 explains the histogram of SLC and MLC threshold voltage where the distribution is a gaussian distribution as previous study confirmed[1]. Figure 2.2(a) shows that for an SLC NAND flash, there only one bit can be extracted from a cell i.e. either 0 or 1 based on its threshold voltage. Figure 2.2(b) is for MLC; 2 bits can be extracted from each cell based on the threshold level. In both cases, the x-axis is the threshold voltage range and the y-axis define the number cells at each voltage level.

2.2.2 NAND flash array:

NAND flash memory is an array of memory cells. This array is divided into blocks. Depending on the flash memory topology, each block will have the cells of each bitline
connected in parallel or in series. Figure 2.3 shows the high-level organization of the flash memory block. Each of the blocks in the NAND flash memory contains multiple rows of cells and each row of cells is connected by a common wordline. All of the cells that are connected by the wordlines are responsible to form a page. For MLC NAND flash memory, the MSBs of all cells on the same wordline are combined to form the MSB page and LSBs of all cells are combined to form the LSB page. Also, within a NAND flash block, all cells in the same column are connected in series to form a bitline (BL). All the cells in a BL have the same ground on one end and a common sense amplifier on other ends for reading data from a cell. To control the BL operations, there is one string select line (SSL) and one ground select line (GSL) transistor of each BL on or off. To enable the operation of a bitline, SSL transistor is used and to connect the BL to the ground, GSL transistor is used.

Figure 2.3 The hierarchical storage in NAND flash array consisting of kilo-bytes of memory cells and the WL electrically connects those cells (called a page of information). Each block consists of multiple WLs. The select gate transistors can be standard MOSFET or FG transistors, depending on manufacturer or technology node.
2.2.3 Read, program and erase operations:

The basic read operation can be performed to a NAND flash memory by applying a read reference voltage to the control gate. This reference voltage senses the cell’s threshold voltage and finds whether the cell is in programmed state or in erase state. To read an SLC cell, it is only required to distinguish between the erase state i.e. 1 and program state i.e. 0. And this requires only a single read reference voltage. For an MLC chip, for LSB page read it only requires distinguishing for the LSBs threshold i.e. Er and P1 and to read the MSB pages it requires checking two threshold references to distinguish between 0 and 1 because of their different distribution. Figure 2.4(a) shows the basic read operation organization of a bitline. In a BL string, the target cell which is to be read is set to a reference voltage by applying the Vread to the control gate and to propagate the sensing value to the sense amplifier, all other cells connected to the BL is set to a pass voltage

![Figure 2.4 NAND flash basic operations. Applied voltage to bitline to (a) read, (b) program, and (c) erase.](image-url)
which turns them on. In this way, voltage is sensed, and data is determined as 0 or 1. This pass voltage is the maximum possible threshold voltage for which the drain and the source are connected regardless of the voltage of the floating gate.

A program operation is almost like a read operation where the program voltage is applied to the control gate. The threshold voltage of a floating gate transistor depends on the charge stored on the floating gate. And the amount of charge depends on the injection and ejection of electrons through the tunnel oxide of the transistor, which is enabled by the FN tunneling effect [2]. During a program operation, electrons are injected into the floating gate from the substrate by applying a high program voltage to the control gate (see Figure 2.4(b)). All other cells in the same BL set to a pass voltage and turned them on. This program operation is done by repeatedly pulsing the programming voltage, which is known as incremental step-pulse programming (ISSP)[3], [4]. After each of the program pulse, a verify voltage is applied to check whether the cell meets the required threshold voltage level or not. The read and program operation takes place on a page-level granularity.

In flash memory, a data overwrite is not permitted. Reprogramming is possible only after an erase operation because ISSP can only increase the voltage of the cell. The erase operation is the one that resets the threshold to the Er state. Figure 2.4(c) shows the erase operation of a NAND block. A high voltage is applied to the substrate so that the electrons from the floating gate can be ejected to the substrate and make the cell erased. The erase operation granularity is block. This operation is performed in a block-wise manner and it’s a costly operation because all cells in a flash block share a common transistor substrate[5], [6].
2.2.4 3D NAND flash memory:

Flash memory technology is going through a transition from planar to three-dimensional (3-D) architecture which promises to extend the incredible growth of bit-density over the next decade. 3D NAND has a set of features that enable it to replace HDD and the most prominent feature is its high-density bit capacity. 3D NAND is stacking more and more layers with time which makes this device highly dense memory. Starting from a 32-layer stacked structure now it is becoming 48, 64, and 96 layers in a single SSD. As layer increases rapidly so do bit density.

Figure 2.5 (a) Planner flash memory cell vs (b) 3-D cell geometry which is a gate-all-around vertical channel MOSFET.

However, the 3D NAND technology (Figure 2.5) is significantly different from its 2-D counterpart in terms of cell structure (planar vs cylindrical cell), material (c-Si vs poly-Si as channel) and fabrication processing (“channel-first” vs “channel-last”[7]. 3-D NAND has a layered structure of alternate metal and oxide. Each metal layer is also called wordline and it acts as the transistor gate. Each memory block consists of a fixed number of memory pages. Memory-read and program operations are performed at the page granularity, while erase is performed at the block granularity.
Chapter 3: Security and Privacy of Scrubbed NAND Flash Memory

Digital sanitization of flash based non-volatile memory system is a well-researched topic. Since flash memory cell holds information in the analog threshold voltage, flash cell may hold the imprints of previously written data even after digital sanitization. In this chapter, we show that data is partially or completely recoverable from the flash media sanitized with “scrubbing” based technique, which is a popular technique for page deletion in NAND flash. We find that adversary may utilize the data retention property of the memory cells for recovering the deleted data using standard digital interfaces with the memory. We demonstrate data recovery from commercial flash memory chip, sanitized with scrubbing, by using partial erase operation on the chip. Our results show that analog scrubbing is needed to securely delete information in flash system. We propose and implement analog scrubbing using partial program operation based on the file creation time information.

3.1 Introduction

Secure deletion of obsolete data from the storage medium is a topic of paramount importance to ensure the privacy and security of the data owner. According to the Data Protection Act (DPA) 2018, the deletion of information must be real i.e. the content should not be recoverable in any way. However, achieving true deletion of user data from the physical storage medium is not always straightforward and it depends critically on the analog characteristics of the specific non-volatile storage elements.
In this chapter, we evaluate the secure deletion concept in NAND flash based non-volatile storage system. NAND flash memory finds a ubiquitous place in today’s computing and storage landscape. Flash memory is widely used in personal electronic gadgets including smartphones, solid state drives, laptops, tablets, USB memory sticks, SD memory cards, etc. Due to the increasing popularity of flash as non-volatile storage media, the concept of secure deletion or sanitization of flash media is getting even more important.

NAND flash exhibits certain unique challenges for secure deletion due to its special characteristics. First, write operation in flash takes place at page level granularity while erase operation happens at block level which requires all the pages in a block to be deleted at the same time. Because of the mismatch in granularity between erase operations and program operations in flash, in-place update of a page is very resource expensive. Second, NAND flash requires erase-before-write constraint, which makes overwriting operation very unfriendly, given the mismatch in granularity between erase and write operation. Third, NAND flash has finite endurance meaning only a fixed number of program and erase operation are allowed on a NAND block. Thus, the flash controller is typically designed to minimize erase operation and ensure wear-leveling of all the memory blocks. In other words, erasing a block for in-place update of a page is not a common practice.

Since in-place updates are not possible in NAND flash, the standard overwrite-based erasure techniques, typically used for hard drives, does not work properly for NAND storage system. Instead, NAND storage usually perform logical sanitization (i.e., the data is not retrievable via the SATA or SCSI interface) by invalidating the page address of obsolete data. The page address mapping in NAND storage is handled by an intermediate firmware layer called Flash Translation Layer (FTL), which performs one-to-one mapping
between logical page address and the physical memory address of the flash media. Thus, for any page update operation, FTL will write the new contents to another physical page (or sector) location and update the address table map so that the new data appears at the target logical address. As a result, the old version of the data remains in the physical storage medium, which can be retrieved by the adversary with advanced memory interfaces.

In order to achieve page-level deletion in flash based storage, the idea of “data scrubbing” was proposed by Wei et al[8]. The key concept behind “scrubbing” based sanitization is the creation of an all-zero page (or all cells programmed), which is equivalent to deletion of data from that page. Since it is not possible to remove charge from the floating gates at page level granularity, “scrubbing” provides an alternative route to digital sanitization by programming all the cells in the page. However, in this chapter, we show that the deleted data is partially or completely recoverable after “scrubbing” due to the analog property of the programmed cells. More specifically, programmed cells in flash continuously lose charge due to fundamental data retention characteristics. As a result, the zero bits (or programmed bits) in the original data loses a portion of the stored charge at the time of erase operation. We call these zero bits as weak zeros since they have slightly lower threshold voltage compared to the freshly written bits. During “scrubbing” a new set of zero bits are created by newly programming the erased cells in the original data. We label the freshly written zeros as “strong zeros” because they have higher threshold voltage compared to the original zero bits (weak zeros) in the data. Thus, careful analysis of the analog threshold voltage of the memory cells in a scrubbed page will reveal the original data.

Our key contributions in this chapter are as follows:
1) We demonstrate that data is partially or fully recoverable from “scrubbing” based deleted page. We use partial erase operation on a “scrubbed” page to recover the deleted data.

2) We find that fundamental data retention (or charge loss) characteristics of flash cells should be taken into account during “scrubbing” to ensure true deletion of data.

3) We propose a new analog “scrubbing” technique in order to make sure data remains unrecoverable after deletion. The proposed technique utilizes the time difference between write and erase operation in order to program the erased bits. This will minimize the threshold voltage differences among the cells in the erased page.

3.2 Background

In this section, we will describe the fundamentals of flash memory cell, its operation, and NAND flash system design.

3.2.1 Flash memory organization and operations:

Flash cell: Figure 3.1(a) shows the device structure of a flash memory cell, which is essentially made of floating gate MOSFET (Metal Oxide Semiconductor Field Effect Transistor). Electrons placed on a floating gate are trapped because the floating gate is isolated electrically from the control gate and the transistor channel by blocking oxide and tunnel oxide respectively. Thus, a flash memory cell stores information in the form of charges (electrons) for an extended period of time without requiring any power supply.

NAND array: Flash memory is organized as two-dimensional arrays of floating gate transistors. A number of cells connected in series, in a column, form a string (see
Figure 2.3) as described in section 2.2.2, which is electrically connected to the metal bit line at one end and grounded at the other end. Cells in a row are electrically connected through a metal Word Line (WL) and constitute a page. The size of a page varies from 2-16K byte depending on manufacturer. There can be multiple pages per metal wordline depending on the storage technology. The number of bits per cell depends on the type of flash chip like SLC (single-level cell or 1 bit/cell), MLC (multi-level cell, 2bits/cell) etc.

Threshold voltage distribution: The threshold voltage ($V_t$) of a flash memory cell varies in analog-way depending on the amount of charge on the floating gate. Due to process variation, there is a cell-to-cell difference in threshold voltage, even though the cells are at programmed or erase state. Thus, program or erase state does not represent a single value of $V_t$. Instead, each state is represented by a $V_t$-distribution. Flash manufacturer generally keep enough voltage margin between the erase state $V_t$ and the program state $V_t$ so that they can be digitized accurately with a single reference voltage. Interestingly, the $V_t$ distribution can be measured with standard digital interface by

Figure 3.1 (a) A floating gate (FG) NAND flash memory cell which stores information in the form of charge on the FG. Metal word-line (WL) act as the control gate of the FG transistor. Charge is injected on the FG through tunneling of electron from Si-channel to FG. Blocking oxide prevents back tunneling of electron to control gate. (b) Typical histogram representation of threshold voltage for erase state cell and programmed state cell.
measuring bit error rate with shifted reference read level (Read Retry operation). The detailed $V_t$ measurement procedure is discussed by Cai et al [9].

Memory operation: Flash memory offers three basic operations: erase, program, and read which are already described in section 2.2.3. Among the program and read, operations take place on a page by page basis, while erase operation takes place on a block by block basis. During a program operation, a high voltage is applied on the WL which acts as a control gate of the MOSFET and attracts channel electrons into the floating gate by Fowler-Nordheim tunneling through the tunnel oxide. These trapped (negative) electrons increase the threshold voltage of the transistor. In erase operation, these trapped charges are removed from the floating gate by the application of high positive voltage on the substrate and the control gate is grounded. The erase state of flash cell represents logic “1”. The programmed state has higher threshold voltage due to the presence of negative charges and it represents logic “0”. Read operation involves sensing the threshold voltage of the flash cell by monitoring the current conduction. During a read operation, logic “0” & “1” are sensed by applying an intermediate (or reference) voltage to the control gate, which is less than the programmed threshold voltage. If the cell does not conduct current at the reference voltage, it is treated as a programmed cell or logic state “0”. If the channel conducts current at the reference voltage, then the flash cell is considered in erase state and represents logic state “1”.

3.2.2 Flash translation layer:

In order to efficiently manage the NAND array’s special characteristics, a firmware layer called flash translation layer (FTL) [10]–[12] is typically used by the storage system which interfaces the host file system with the raw NAND memory. FTL provides a block
access interface to the host file system by mapping the logical addresses in block layer to physical addresses in NAND flash. In addition, FTL contains firmware module for garbage collection and wear leveling[10]–[12]. The garbage collection module periodically reclaims all the invalid pages in the media in order to perform block erase operation, which will free-up memory space for new data. The wear-leveling module manages the limited endurance of the flash media by ensuring uniform program-erase operation on all the blocks.

3.3 Threat model and assumptions

Adversary Model: We assume the adversary has one-time access to the flash memory device. In addition, we assume that adversary can perform multiple read and erase operations on the content of the flash. We also assume that adversary aims to illegitimately derive sensitive information which is not available through a “legitimate” interface. For example, we assume adversary has access to the raw NAND memory chip and he/she can perform low level memory operation, such as partial erase, shifted read or read retry, etc.
Assumption: We assume adversary can read the data from the NAND flash without any error correction. Most of the NAND chips do not include error correction engine on the memory chip. Instead, the ECC engine is typically included in the FTL. We also assume that adversary can access the NAND flash chip with bypassing the FTL.

3.4 Data retrieval after scrubbing

3.4.1 Data retention (DR):

Flash memory technology has finite data retention characteristics as the stored charge on the floating gate (and trapped electrons in the oxides) continuously leaks through surrounding oxides [13]–[15]. Because of DR, when data is stored and kept for some time, the programmed state cell tends to lose its charge and its threshold becomes lower. Figure 3.3 describes the data retention effects on the $V_t$ distribution for SLC type of storage using histogram representation. In Figure 3.3(a), there are two states: erase state and programmed state. Flash manufacturers keep enough voltage margin between the two states. The read reference voltage is typically chosen in the middle of the voltage margin. When the cell voltage is greater than the reference voltage, it reads as logic zero and when the cell voltage
is lower than this, the data is read as one. In Figure 3.3(b) we show the DR effect on the cell $V_t$ distribution. Usually, program state $V_t$ distribution moves down with DR, while erase state $V_t$ distribution remains almost the same. If the data retention time is not large (less than 1-2 years), the program state will still be read as zero even after down-shift of cell $V_t$ distribution.

3.4.2 Scrubbing after data retention:

When data scrubbing is applied for page level sanitization after DR, all the data of the page becomes zero. It is important to note that even though all the bits are read as zero digitally, their threshold voltage distribution will have significant and detectable differences. In Figure 3.3(c), the data retention effect on the “scrubbing” process is explained. When scrubbing operation is applied to a page, which has gone through a finite time of data retention, only the erased cells are programmed. The zeros in the original data remain at the same $V_t$. However, the newly written zero has higher threshold than the old zeros as old zeros already lose some of their charges. In Figure 3.3(c), the red zeros are the old zeros which have a lower threshold distribution than the black zeros which have a higher threshold distribution. Thus, the data retention property is the key characteristic to identify the same logical zero as weak zero and strong zero based on their threshold voltage distribution. In other words, if scrubbing is done immediately after writing the data, the difference between old and new zeros will be minimal and it will be difficult to recover the data. If the time difference between write and scrubbing is high, there will be a higher chance that data will be recoverable.

3.4.3 Partial erase:
We utilized partial erase of a block in order to recover the scrubbing based deleted data. A full erase is the process where all the bits in a memory block turn into the logic state “1”. The datasheet of the specific flash chip reports the typical erase time of a block. If the erase operation is interrupted in between by issuing a “RESET” command, then the operation is called partial erase. Partial erase will lower down the $V_t$ distribution of the programmed cells in step by step, which provides a method to distinguish the strong vs weak zeros using standard digital interfaces. Alternative methods exist in order to determine the cell $V_t$ by using digital interfaces, such as read retry, which involves counting fail bits with shifted read levels[16], [17]. However, many SLC NAND chips do not include read retry feature. Hence partial erase offers a more generic method to distinguish weak vs strong zeros.

3.4.4 Data recovery with partial erase:

The adversary can utilize the partial erase operation to distinguish between strong and weak zeros on a fully scrubbed page (meaning all the data being zeros). For example, if the adversary performs partial erase on a fully scrubbed page with fine resolution of erase time, the weak zeros will first turn into ones while the strong zeros will still be read as zeros. Thus, adversary can recover the original data by distinguishing the strong and weak zeros on a deleted page.

In Figure 3.4 we illustrate the data recovery process with an Einstein image (binary) as an example. We first store the binary image in a NAND block (Toshiba SLC Part # TC58NVG3S0F). The size of the image is 276,000 bytes and it requires 64 SLC pages for getting stored in the memory. In order to accelerate the data retention effects, we then bake
the NAND chip at high temperature (120°C) for 3 hours and read back the image data. Figure 3.4(b) shows the post-bake Einstein image. From a digital viewpoint, there is no difference between the pre-bake Einstein image and the post-bake one.

---

Algorithm 1: Partial erase on scrubbed data

**Initialize:**

- Target block where data has been stored and scrubbed previously \((\text{TargetBlock})\);
- Measured partial erase time based on data retention information \((\text{PE}_{\text{time}})\) and define delay \((t_{\text{d}})\);
- Number of pages in a single block \((\text{Block}_{\text{page}})\);

**Perform:**

- Retrieve data from each page of target block using partial erase \((\text{Retrieve}_{\text{page}})\);

1: Issue NAND block erase command;

2: Apply time delay \((t_{\text{d}})\);

3: Issue \(\text{RESET}\) command \((\text{FFh})\);

4: Issue \(\text{READ}\) command to read NAND flash data;

5: Save the retrieved bytes;

---

However, the analog threshold voltage of the memory bits holding the data in Figure 3.4(a) and the Figure 3.4(b) are distinctively different as illustrated with the
Figure 3.4 Data recovery from scrubbed pages. (a) Original Einstein image (460 × 600) which is stored to the NAND flash. (b) Original image after data retention takes place (c) Scrubbed image, this is all ‘0’ image. We bake the chip for 3 hours in an oven of temperature 120°C in order to accelerate the data retention effect. (d) The raw image recovered using partial erase operation. (e), (f), (g) & (h) Histogram representation of threshold voltage for corresponding image of (a), (b), (c) & (d).

downshifted threshold voltage distribution of zero bits in Figure 3.4(f). Next, we perform the scrubbing-based deletion operation and read the data back. Figure 3.4(c) shows the deleted image. As expected from a scrubbed NAND data, the image looks completely black or all the bits of the image are at zero state. However, in terms of analog threshold voltage, there is an important distinction between the zero bits. The original zero bits have slightly lower threshold voltage than the newly created zero bits, even though digitally both are read as zeros. Finally, we perform partial erase operation on the scrubbed image to recover back the original data. Partial erase operation will shift down the $V_t$ distribution of both strong and weak zeros in such a way that the memory read operation with a fixed reference voltage will identify most of the weak zeros as one bits and most of the strong zeros as zero bits (see Figure 3.4(h)). We invert the bit map after partial erase operation and plot the recovered image in Figure 3.4(d).
Algorithm 1 briefly describes the command sequence used for data retrieval process. First, we select a block where data was stored for some time and then scrubbed recently. So, all the data is read as 0 with standard NAND read command. Then, according to the data storing information (i.e. the time when the data stored and scrubbed), we calculate the optimal partial erase time ($PE_{time}$). And, then upon applying the precise time delay for the partial erase process, we also issue the NAND Read operation to read data from a specified page. The data read after the partial erase is essentially the inverted version of original stored data. So, we invert the data and save as $Retrieve_{page}$ data. And the final stopping criteria requires 90% of the data become in the erased state. Until this requirement fulfills, the partial operation will continue for the specific block and keep saving data from the specified page for each partial erase operation.

3.5 Analog scrubbing

In principle, analog sanitization of the flash media will ensure true destruction of the stored data. Analog sanitization of semiconductor memory is always challenging because there are many electronic processes that leave imprints of remnant data on the device characteristics[18]–[20]. The block erase operation of the NAND flash device is closer to the analog sanitization of the flash media, as it ensures information is lost by removing the floating gate charge from the programmed cells. However, due to fundamental array architecture of NAND flash, there is no equivalent page-level erase command that converts all the bits in a page into erased bits. Hence developing analog scrubbing method is essential in order to securely delete page data in the NAND array. In this work, we propose the analog sanitization method of NAND memory pages using the history of data creation such as page creation time. The basic idea is to create an all-zero
page (similar to digital scrubbing) with the additional constraint that all the zero bits have undistinguishable analog threshold voltage distribution. We implement this idea using partial program technique as described in the next section.

3.5.1 Partial page program:

A NAND page generally takes \(~100-1000\) µs to be fully programmed based on different technology. The partial page program method on a NAND page is typically implemented by issuing a NAND RESET command after the NAND write command. The RESET command will forcibly stop the NAND write operation before its stipulated time. As a result, the memory cells get programmed to a lower threshold voltage level than the corresponding fully programed threshold level. In addition, introducing a time delay \((t^p_d)\) between the NAND write command and the RESET command, it is possible control the analog threshold voltage of the partially programmed cells.

3.5.2 Analog scrubbing with partial page program:

The goal of analog scrubbing is to match the threshold voltage distribution of the original zero bits and the newly created zero bits during page scrubbing. The challenge here is to estimate the partial program time during scrubbing which depends on the mean threshold voltage value of the original zeros of the page. Hence the knowledge of page creation time and an accurate model for data retention characteristics of the memory chip will be critical to implement this method.

Algorithm 2 explains the process of analog scrubbing, where we estimate the program time for analog scrubbing process based on data retention information. Note that NAND flash has a default page program time \(t_{PROG}\) of \(~100-1000\) µs.
Algorithm 2: Analog scrubbing with partial program

Initialize:

A randomly selected valid block where data has been stored previously;

A randomly selected page in the selected block;

Estimated approximate time delay for analog scrubbing based on data retention information ($t_{dp}$);

Flash chip page program time ($t_{pro}$);

Perform:

Make all the data either strong zero or weak zero based on time delay;

1: Issue NAND page write command;

2: Apply time delay ($t_{dp}$);

3: Issue RESET command (FFh);

4: Issue READ command to read NAND flash data;

5: If scrubbed data < 97% then

6: Repeat 1 to 5

We first select a block where some data has been stored previously and select a page to be read. In this method, depending on the data retention information, program time
delay $t_{d}^{P}$ is defined for a page ($t_{d}^{P} \leq t_{PROG}$). Then we issue the NAND write operation to implement the analog scrubbing. In this case, some of the bit might not be programmed, so if the percent of bit programmed is less than 97%, this process will take place again until the criteria fulfills. After performing scrubbing in this way, difference between 0’s threshold voltage distribution is not noticeable, and an adversary is not able to recover the data from this page fully or partially.

3.6 Implementation and evaluation

<table>
<thead>
<tr>
<th>Part #</th>
<th>Manufacturer</th>
<th>Block erase time ($t_{ER}$)</th>
<th>RESET time (Erase)</th>
<th>Bit recovery efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC58NVG3S0F</td>
<td>Toshiba</td>
<td>3ms</td>
<td>500µs</td>
<td>77.54%</td>
</tr>
<tr>
<td>MT29F8G08ABACA</td>
<td>Micron</td>
<td>2ms</td>
<td>500µs</td>
<td>53.72%</td>
</tr>
<tr>
<td>K9F2G08X0A</td>
<td>Samsung</td>
<td>1.5ms</td>
<td>500µs</td>
<td>-</td>
</tr>
<tr>
<td>MT29F4G08ABADA</td>
<td>Micron</td>
<td>700µs</td>
<td>500µs</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3.1 Evaluation of bit recovery efficiency on different NAND chips.

3.6.1 Experimental set-up:

A custom design hardware board is used in order to interface the commercial off the shelf flash chips with the computer. The board contains a socket to hold the flash chip under test and an FT2232H (Future Technology Devices International Ltd) break-out board for USB communication. For the evaluation purpose, we have used SLC NAND flash memory chips from different flash manufacturers including Toshiba, Micron, and Samsung. The exact part number for all the chips used is listed in Table 3.1.

3.6.2 Data retrieval efficiency after scrubbing:

We evaluate the data recovery efficiency from a digitally sanitized all zero page in Figure 3.5. The key parameter in the data recovery process is the precise control of the
partial erase time. If the duration of erasure is kept long, most of the bits will be erased (similar to standard block erase operation). On the flip side, if the erase duration is too small, then most of the bits will remain in zero states. Figure 3.5(a) illustrates the gradual data recovery process as a function of erase duration. For a clear illustration, we plot the impact of partial erase time on strong vs weak zeros separately in Figure 3.5(b). For complete recovery of the scrubbed image, it is required that all the weak zeros (zero bits of the original image) are converted to ones while all the strong zeros (one bits of the original image) remain at zero state. Due to overlap in the $V_t$ distribution of the strong vs weak zeros, a partial recovery of the original image is possible in practice. For example, in the Figure 3.5(b) we found that at the beginning of erase operation weak zeros turned into ones much faster than the strong zeros. However, a significant percentage of strong zeros also flip to ones before all the weak zeros are flipped. Hence there is a narrow window of opportunity for the partial erase duration that ensures recovery of most of the original data.

![Graph](image1.png)

**Figure 3.5** (a) Illustration of bit accuracy of a recovered image vs partial erase time. The image was scrubbed with zero-overwrite method after 3 hours of bake at 120°C. (b) For clear illustration, we plot the percentage of weak zeros that flipped into ones as well as the percentage of strong zeros that remained at zero state with respect to erase time.
We define bit accuracy as a new metric for recovered image which measures the percentage of correctly identified bits of the original image data after partial erase operation. We find that there is an optimum erase duration that gives the highest bit accuracy. Please note that a completely scrubbed image (all zero bits or all one bits) will also have a bit accuracy approximately 50% assuming equal number of zeros and ones in the image. However, in order to recognize an image, the correctly identified positions also play a significant role. Hence in Figure 3.5(a) we find that even though the bit accuracy of the recovered image is low for certain erase duration, the image is correctly recognizable.

The other important point to note here is that the maximum bit accuracy is a function of data retention time or the storage history of the image. Typically, the longer the duration of high temperature bake in our experiment (or older the stored data), the separation between $V_t$ distribution of strong vs weak zeros will be wider. This will increase the bit accuracy of the recovered image. In Figure 3.6 we plot the maximum bit accuracy of the recovered image for different high temperature bake time. We find that the longer the bake

![Figure 3.6 Bit recovery accuracy versus bake time for stored Einstein image on two different NAND flash chips.](image)
duration, the higher the bit accuracy. Note that the maximum bit accuracy of the recovered image corresponding to different bake time is a function different partial erase duration.

We have performed the evaluation of a partial erase based image recovery on chips from different flash manufacturers and found that the partial erase method works for those chips which have longer block erase time. The block erase time of NAND flash chip is defined in the datasheet by the manufacturer i.e. $t_{ER}$ and typically $t_{ER}$ varies from ~1-10 ms. The longer the block erase time, it is easier to control the partial erase operation using the digital interface. Note that the partial erase operation is implemented using our measurement set-up as follows: we issue a block erase operation for a specified NAND block and then we issue the NAND RESET (FFh) command after certain pre-defined time delay ($t_d^e$). The RESET command takes a finite time (few hundreds of micro-seconds) to forcibly terminate the erase operation. In addition, the effective time for the pre-mature termination of erase operation depends on the delays associated with issuing commands by the digital interface. Hence the minimum value of the partial erase duration is limited by the time corresponding to the RESET command and the delays associated with the digital interface. For some of the chips as listed in Table 1, we found that the minimum value of partial erase duration is comparable to the block erase time, and hence the NAND block gets completely erased after the partial erase operation with $t_d^e = 0s$. Thus, the data recovery process could not be successfully implemented on those chips using our experimental set-up. A faster digital interface is needed in order to implement data recovery algorithm on those chips with lower block erase time.

3.6.3 Effectiveness of analog scrubbing:
In order to show the effectiveness of the proposed analog scrubbing, we perform the following step-by-step experiments. First, we write the same Einstein image on a NAND block and bake it for 3 hours at 120°C to emulate the data retention effects. We then create an all zero page using partial programming technique. The partial program duration is calculated based on memory’s data retention characteristics. We then repeat the partial erasure based data retrieval process (as discussed in Section 6.2) on the scrubbed image. The results are shown in Figure 3.7. We find that recovered image is difficult to be identified compared to the fully scrubbed image. These results show the prospect of analog scrubbing technique for the true deletion of data. However, the accurate implementation requires careful characterization of partial program duration as a function of data retention time, which remains a topic for future investigation.

3.7 Related work on NAND sanitization methods

In this section we provide a brief overview of the state-of-the-art sanitization methods for NAND flash memory systems. Since in-place updates are not possible in NAND flash, the standard multiple overwrite-based erasure techniques, typically used for
hard drives, do not work properly for NAND storage system. Instead, following methods are typically employed by NAND controller for sanitization:

Block erase: Block erasure method is a basic NAND command to remove data from all the pages of a NAND block. The method essentially removes electronic charge from the flash cells and hence physically erase the data from the NAND media. Typically, during garbage collection process this method is used to remove old invalid data once the drive is almost full [10]–[12]. Thus, this command is sparingly used by a NAND controller. However, there are certain sanitization proposals which use this command for secure deletion [21]–[23]. The major drawback of block erasure based methods is the poor performance caused by the significant valid data migration overhead [23]. The other bottleneck for the frequent block erasure is the finite endurance limit of NAND flash technology. Thus, block erasure based immediate page deletion techniques are not a practical and efficient solution for NAND storage.

Logical sanitization: Since block erasure methods suffer from poor performance, NAND storage usually performs logical sanitization by invalidating the page address of obsolete data. The page address mapping in NAND storage is handled by FTL, which performs one-to-one mapping between logical page address and the physical memory address of the flash media. Thus, for any page update operation, FTL will write the new contents to another physical page (or sector) location and update the address table map so that the new data appears at the target logical address. As a result, the old version of the data remains in the physical storage medium, which can be retrieved by the adversary.

Encryption based sanitization: Several authors have recently proposed NAND sanitization methods based on an encryption technique [24]–[29]. The basic idea in this
method is to encrypt the user file with an encryption key and store the encrypted data and the key in two separate NAND blocks. Secure deletion is achieved by removing the keys, which can be done efficiently as keys require smaller memory space. Even though encryption based techniques are quite fast, they suffer from the following drawbacks. First, encryption based technique carries the risk of data recovery as its implementation may have certain issues, such as random number generation (for encryption key) which can be compromised by a motivated adversary[30]. Second, encryption based sanitization requires proper sanitization of key storage block and any other derived values that might be useful in cryptanalysis. Third, several existing storage solutions and resource constrained embedded platforms may not include any encryption/decryption module and hence cannot implement this technique.

Scrubbing or zero-overwrite based digital sanitization: In order to achieve page-level deletion in flash based storage, the idea of “data scrubbing” was proposed [8], [23], [31]. The key concept behind “scrubbing” based sanitization is the creation of an all-zero page (or all cells programmed), which is equivalent to the deletion of data from that page. Thus, “scrubbing” provides an alternative route to digital sanitization by programming all the cells in the page. However, we have shown in this chapter that the scrubbed data is partially or completely recoverable due to the analog property of the programmed cells.

History independent erase: Recently, several researchers have proposed secure NAND deletion methods which will not only remove data from the storage medium but also conceal deletion history from the system[32]–[34]. For example, Jia et al.[33] proposed undetectable secure deletion in flash system by using a partial scrubbing technique and removing any structural artifacts of past deletion operation from the flash
system. Similarly, Chen et al. [34] designed HiFlash, a history independent flash device, which will remove all the history related artifacts in the flash layout.

3.8 Conclusions

In this chapter, we show that digitally sanitized (zero overwrite based “scrubbing”) NAND flash storage media still maintains the previously written information in the analog threshold voltage characteristics. The data retention property of the flash memory cells causes difference in the analog threshold voltage of the original zero bits in the data and the newly created zeros during scrubbing. We experimentally demonstrate that the difference in the analog threshold voltage can be exploited to recover the deleted data from a fully scrubbed all-zero page. We utilize partial erase technique to recover the deleted data and our evaluation shows more than 75% bits are recoverable depending on the specification of the NAND chip, memory cell’s data retention characteristics and the nature of the image data. Finally, we describe a new method for analog sanitization of NAND memory pages using page creation time and partial program technique. Our evaluation shows promise of the proposed technique for analog sanitization and true deletion of user data from the flash media.
Chapter 4: Reliability Issues of NAND Flash in Computing

NAND flash memory is a popular choice for storing a large set of model weights of an Artificial Neural Network (ANN) in many Internet of Things devices and edge computing applications. While being used as a weight storage device, the bit error rate of flash memory plays a significant role in the performance of the ANN application given that the error correction engine of the memory controller is turned off due to power constraints. In this chapter, we propose a novel weight storage method in NAND flash memory, which will significantly suppress the effects of bit error rate on the ANN’s model weights and its performance. The proposed weight storage method utilizes the NAND flash’s page-to-page variability in favor of storing critical bits of the model weights on more reliable pages. In order to demonstrate the benefit of the proposed method, we perform different reliability experiments on commercial flash memory chips containing the trained weight values of an ANN application. The experimental evaluation shows that the proposed method outperforms the traditional weight storage method and ensures prediction accuracy more than 90% even with a bit error rate exceeding 1% value. We also show the effectiveness of a multi-level-cell 3-D NAND flash chip as a weight storage device for a neuromorphic computing system under radiation environment. We find that the error-correction codes can be avoided for storing model weights in 3-D NAND for enabling low-power computing applications without sacrificing much accuracy (radiation dose <10k rad). Additionally, radiation induced BER data shows layer-to-layer variations, which can be utilized in favor of improving neural network’s accuracy.
4.1 Introduction

Artificial neural network (ANN) based applications are becoming increasingly important in many Internet of Things (IoT) devices and edge computing platforms. There are several emerging applications such as self-driving cars, robotics, drones that require in-place processing of large input data using the ANN based accelerators [35]–[37]. The most common approach for implementing the ANN accelerators on the resource-constraint IoT devices is the off-chip training of the network and the storage of the trained model weights in the non-volatile memories on the IoT platform. One key constraint with this approach is the storage of a large number of model weights, which may require more than giga-bytes of storage space depending on applications [38].

Several non-volatile memories are being evaluated for storing ANN model weights including magnetic tunnel junction-based magneto-resistive random access memory (MRAM), resistive random access memory (RRAM), phase change memories (PCRAM) and flash memory [39]–[47]. Each memory technology offers certain advantages and disadvantages. For example, MRAM technology is fast but it can store only one bit per cell due to limited magneto-resistance change between ON and OFF state. RRAM offers a higher resistance range, but the technology is not mature yet. Furthermore, robust multi-level programming is found to be challenging for RRAM and PCRAM due to low controllability of the filament formation and heat diffusion. In contrast, flash memory is a mature technology which can easily store multiple bits per cell offering high bit density at low cost. Hence flash memory is an attractive choice for storing a large number of model weights in the edge devices for neural network applications.

Several recent works have highlighted the usefulness of flash memory technology
for storing the large number of neural network weights[46]–[49]. However, a key concern for using flash technology as weight storage media is its fundamental reliability constraints, including data retention (DR), read disturb (RD) and endurance[12], [13], [15], [50]–[52]. Also, it has reliability concerns regarding radiation, including total ionization dose (TID) and single event effects (SEEs) [53]–[58]. Reliability issues cause data corruption, which is quantified as bit error rate (BER) in the stored data [57], [59], [60]. Hence standard flash controller uses error correction codes (ECC) in order to manage the bit error rate during memory read. The ECC engine usually demands high power and increases overall complexity during memory-read which can be detrimental in the low-power edge devices and internet of things (IoT) applications.

The key objective of this chapter is to determine the accuracy of an ANN against the bit error rate of NAND flash which is used as a weight storage device without the ECC engine. We investigate the impact of inherent flash reliability, such as endurance, and data retention on the stored weight values of an ANN that was trained off-line to identify handwritten numerical digits using MNIST dataset. For the experimental evaluation, we have used commercial-off-the-shelf (COTS), multi-level cell (MLC) NAND flash memory chip of size 64Gb from Micron technology. We first store the trained model weights in the chip and perform different reliability experiments, such as data retention using high temperature bake and endurance test using repeated program-erase stress on the chip. In order to evaluate the impact of reliability effects, we measure the raw bit error rate (BER) on the stored model weights by disabling the ECC engine. We then evaluate the ANN accuracy with the BER affected model weights. The key findings of this chapter are as follows:

1) We find that the prediction accuracy of a neural network gets significantly
affected when the bit error rate exceeds 1%. In addition, we observe significant fluctuation in the prediction accuracy for BER >1%.

2) We also irradiated using X-rays, which introduces BER in the stored weights. We find that the radiation tolerance of ANN strongly depends on the type of flash memory where the model weights are stored. Experimental evaluation shows that recognition accuracy of the ANN remains robust (>97.5%) for TID ~ 10 krad in 3-D NAND storage, whereas it decreases rapidly in case of 2-D NAND storage (accuracy < 95% for TID ~1 krad).

3) We show that the performance degradation of ANN with NAND BER can be explained with random error injection with uniform probability distribution.

4) Finally, we propose a new weight storage concept in the NAND flash which ensures that the stored weights are least affected by NAND reliability effects. We experimentally demonstrate that the proposed method will offer higher accuracy compared to the traditional method of using NAND flash as a weight storage device.

4.2 Artificial neural network implementation

We first perform off-chip training of the ANN for identifying hand-written numerical digits. The MNIST dataset is used to train and test the ANN. For training purposes, we use 60,000 handwritten data and for testing 10,000 handwritten data is used from the MNIST dataset. The model ANN, which is shown in Figure 4.1(a) has 784 input attributes corresponding to the input image of size 28×28 pixels. The ANN has 10 output attributes corresponding to each numerical digit and we define 250 hidden nodes in between input attribute and output class. We obtain a total of 198,760 model weights from the network. We stop the network training when we achieve training accuracy of 99.95%.
This network requires almost 52 hours to complete the training using an Intel Core i7-8550U processor with 12GB RAM based workstation. The probability density function of trained weight values (normalized) is shown in Figure 4.1(b). We find that most of the weights are distributed near zero. In order to store the trained weight values in the external memory device, we express the trained weights as 8-bit fixed-point data format. The most significant bit (MSB) is the sign bit and the rest of the bits are for representing the numerical value. For example, the weight -0.515625 is represented in 8-bit fixed point data as 11000010. Due to this digitization of weights, we find that the implemented neural network occasionally gives false identification for few test images as illustrated in Figure 4.1(c). Figure 4.1(c) shows an example of input and output digits where we see that digit 4
(a particular test case) is not recognized properly by the trained network. We count the number of such false identification from the 10,000 MNIST test images and find 97.64% as recognition accuracy of the trained network when we use the actual weights and 97.56% as recognition accuracy of the same network when the weights are compromised to 8-bit fixed point.

Figure 4.2 Experimental setup to interface NAND flash memory with PC.

4.3 Experimental evaluation of NAND reliability impacts on ANN performance

4.3.1 Details of NAND flash chip and measurement setup:

In order to emulate the real-life ANN applications, we first store the off-chip trained weight values on commercial off the shelf NAND flash memory chips. For this evaluation, we have used NAND chips, with multi-level-cell (MLC or 2-bit per cell) storage, from Micron Technology. The part number for 3D chip is MT29F256G08CBCBBWP-10:B (32-layer 3-D NAND), while part number for the 2D chip is MT29F64G08CBABAWP:B TR (20 nm planar technology). In order to interface the raw NAND chip with the computer,
we have used a custom designed hardware board (see Figure 4.2). The board includes a socket to insert the NAND flash chip and an FT2232H mini-module from Future Technology Devices International (FTDI) to interface the memory chip with computer through USB connection. The FT2232H mini-module enables USB to UART interface with high speed (480Mb/s). The hardware setup allows us to access the raw memory bits without any error correction. All standard NAND flash based memory solution such as USB drive, solid state drives, and SD card performs error correction in the memory controller in order to manage the NAND reliability. TID tolerance of the flash chip was evaluated using X-ray irradiation and it was performed using a JewelBox-70T (Glenbrook Technologies) 60-keV X-ray system with a dose rate of 2k rad (SiO$_2$) / hour. All terminals of the device under test (DUT) were grounded during exposure.

In order to store the trained weight values in the NAND flash, we first express the trained weights as 8-bit fixed-point data format and store them without error correction codes (ECC). The most significant bit (MSB) is the sign bit and the rest of the bits are for representing the numerical value. The required storage space in the flash is 198,760 bytes since the number of model weights in our ANN implementation is 198,760. All the weight values are stored in one memory block of the chip. The memory block consists of 256 pages, each page size being 8k bytes for 2D and 1024 pages for 3D.

4.3.2 Effect of bit error rate of NAND flash on stored weights:

We evaluate the impact of NAND flash BER on the stored weights by performing reliability experiments on the MLC chip. More specifically, we perform data retention test and endurance effect evaluation. Data retention test is performed by baking the chip at high temperature. The endurance evaluation of memory blocks is done by repeated program
erase operation (or PE cycles) with all zero data pattern. In the following, we provide the step by step operations that were performed to quantify the impact of NAND BER on the stored weights and the performance of the ANN.

1) We store the model weights on a number of memory blocks in the chip with different endurance conditions. For example, we choose four memory blocks with PE cycle count 0, 100, 500, and 1000 respectively. The block with PE cycle count zero implies a fresh block.

2) Immediately after writing the weights in the NAND flash, we read back the digitized weight values. We observe few bit errors in the read back data. The read errors are more on the highly used memory blocks (see Figure 4.3). Since the chip we are using is an MLC chip, it has less tolerance to thermal noise compared to single-level cell (SLC) technology. Hence it shows non-zero BER even after immediately writing on the chip.

Figure 4.3 Reliability issues on stored NN weights in NAND Flash memory based on usage: (a) Measured BER with respect to bake time increases with higher program/erase cycle. This figure illuminates that although higher PE has a significant effect on BER, a 2% usage (max endurance ~ 5000 PE) is quite acceptable in terms of NN weights’ accuracy, (b) Recognition accuracy for MNIST dataset for different memory usage. We find till 100 PE cycle NN accuracy remains acceptable with bake time. However, for PE > 500, data retention effect becomes significant, which conspicuously reduces the detection accuracy of MNIST dataset.
3) Next, we bake the chip at a high temperature for a certain duration. High temperature bake will accelerate the charge loss from the memory bits and hence it will emulate the accelerated data retention. After each bake duration, we read back the stored weights and quantify the BER as shown in Figure 4.3(a).

The increase in BER with the high temperature bake is well understood from the NAND flash reliability literature[14], [51]. Typically, high temperature bake accelerates the charge loss from the flash cells which increases the BER. In addition, more the number of usage (or PE count) of a memory block, higher is the BER. Higher PE operation degrades the quality of the floating gate oxides by creating more trap sites, which in turn increases the BER. Figure 4.3(a) thus confirms all the expected reliability characteristics of NAND flash memory with respect to the stored weights.

Next, we evaluate the impact of induced BER in the model weights on the ANN’s prediction accuracy. We reconstruct the weight values from the binary data obtained by reading the memory. Note that the binary data includes the reliability induced BER and hence the reconstructed weights are not exactly the same to the trained weight values. With the reconstructed weights, we evaluate the ANN’s recognition accuracy using the MNIST data set. Figure 4.3(b) shows the corresponding recognition accuracy value. As expected, the recognition accuracy degrades with higher bake time due to increased BER. In addition, we find that the accuracy degradation is significantly higher for the weights stored in highly used (or more PE cycle count) memory blocks. Please note that a standard error correction engine may correct the data retention induced BER, however, for resource constrained IoT applications error correction for reading the weights will incur significant power
dissipation. Thus, turning-off the ECC engine for flash-based ANN weight storage applications has an important trade-off for power vs reliability.

We also evaluate the impact of TID on the stored model weights by performing X-ray exposure on a previously programmed chip. Ionizing radiation induces BER in the stored model weights as shown in Figure 4.4. We reconstruct the weight values from the binary data obtained by reading the memory which includes the radiation induced errors. Fig. 4.4 shows our experimental result for both 2-D and 3-D NAND where we plot the BER and corresponding recognition accuracy of ANN versus total radiation dose. We find ANN accuracy remains relatively unaffected for TID ~ 10 krad in case of 3-D NAND, while it is significantly degraded in case of 2-D NAND. We note that 2-D NAND has significantly higher BER compared to 3-D NAND, which is the key reason for the accuracy variation between 2-D vs 3-D NAND. Additionally, we observe that unlike 2-D NAND, 3-D NAND shows large BER variation on different word lines. This wordline variation can be utilized further for improving the ANN accuracy. For example, if we store the more critical model weights (or bit positions) in low BER wordlines and vice versa, it will improve ANN accuracy further.
4.3.3 Statistical analysis of ANN’s performance vs BER:

We observe that the recognition accuracy of the ANN fluctuates significantly depending on the exact weights that have been modified due to NAND BER. In order to quantify the statistical fluctuation in the prediction accuracy, we use several memory blocks to store the trained model weights. Interestingly, we find that even if the BER remains the same, the prediction accuracy may differ significantly from one set of stored weights to the other. Even though each memory block stores the same set of weights, the bit error location will differ depending on the physical location of the block. Thus, the reconstructed weights from the binary data from two different memory blocks will be different in the presence of BER. Since ANN performance depends differently on different weight values, we find significant variability in the recognition accuracy for the same BER as illustrated in Figure 4.5(a).

For a deeper understanding of the ANN’s performance variability, we theoretically analyze the recognition accuracy of the ANN by injecting random errors in the digitized

![Figure 4.5 Classification accuracy: (a) Experimental recognition accuracy versus BER while traditional storage method is used to store the NN weights in flash memory (b) Theoretical average recognition accuracy versus BER while error added randomly to neural network weight for MNIST dataset where each data point shows minimum and maximum value for that specific BER.](image)
values of the trained weights. For this evaluation, we express the trained weights with 8-bit fixed point data and save the large bitstream in a Matlab array. We then randomly flip few bits using the random number generator of Matlab. Then, we reconstruct the weights and find the prediction accuracy using the ANN. Figure 4.5(b) shows the theoretical analysis of the recognition accuracy due to the randomly injected errors in the model weights. We find that the mean value of the ANN’s prediction accuracy as a function of BER remains comparable for both randomly injected errors and bit errors caused by NAND reliability.

4.4 New weight storage concept and results

We propose a page-priority based weight storing method to improve the ANN performance in NAND flash. The proposed weight storage method is based on the following two key observations:

1) Even though the model weights are expressed with 8-bit fixed point data, only the first few bit positions from MSB affect the accuracy more significantly. This is
illustrated in Figure 4.6(a) where we show the impact of bit errors in each bit position on the overall recognition accuracy of the ANN. For fair comparison we manually inject random bit errors in each bit position and compare the recognition accuracy for a given BER. Please note that every weight consists of 8-bits, so there is eight bit position for each weight and we inject random errors in each of these bit positions.

2) NAND flash exhibits significant page to page BER variation within a memory block. For example, in MLC NAND each metal word line consists of two shared pages (MSB page and LSB page), which have significantly different reliability characteristics as shown in Figure 4.6(b).

Based on the above observations we propose to store the more sensitive bit positions of the model weights in less error prone flash pages. Figure 4.7(a) illustrates the traditional weight storage method where weight bits are stored sequentially to the flash memory pages.
pages regardless of their priority. Figure 4.7(b) illustrates our priority based storing method where critical weight bits are stored to the low BER pages and less critical weight bits are stored to the relatively high BER pages. For example, we store the first 4 bit positions (from MSB) of all the model weights in 4 different flash pages which were found less error prone. Similarly, the next four bit positions were stored in 4 more pages which have higher BER. While reconstructing the model weights, we read all the 8 pages one by one and form the 8-bit model weights. In general, we need to order the bit positions as mentioned in our first observation and based on the second observation we prioritize the NAND flash pages so that reliability issues can have less effect on the important bits for which accuracy mainly degrades. Then we store the sensitive bits to the less error prone pages. For the reconstruction of weight, we read each page carefully and reorder the bits based on the storage order. Suppose, we have two pages of bits which are on page 6 and page 7. And page 7 shows more error tolerance than page 6. So, we store our most important bits to page 7 and then page 6. While reconstructing the same weights, we read both pages and again sort the bits based on their previous storing order. So, we reorder page 7 first and then page 6 to its next. Then, according to the bit position priority, we consider one bit from a single bit plane and merge the 8 bits from 8 different bit planes to get a single weight. By repeating this procedure, we reconstruct all the weights and perform prediction accuracy to the network. We store NN weights based on the above proposed method and perform the reliability experiment. Finally, we compare the result with the traditional weight storage method. Figure 4.8 shows the experimental results with the new weight storage concept in the NAND flash. We find that the proposed method has two main advantages: (a) We find that the NAND BER has significantly less impact on the overall
ANN’s recognition accuracy. In addition, (b) the uncertainties in the prediction accuracy is also significantly reduced with the proposed method.

4.5 Literature review on in-memory computing

In this section, we provide a brief literature review of on-chip training techniques for neural network implementation. A common approach for hardware implementation of the neural network is biologically inspired synaptic time-dependent plasticity (STDP) using the timing difference between pre and post synapses neuron signals [61]–[63]. This type of synaptic behavior can be imitated by using the analog behavior of NVMs [64]. Silicon synaptic transistor is proposed for a hardware-based spiking neural network [62]. The emerging metal oxide resistive random access memory (ReRAM) is one of the promising memory technology in which crossbar arrays can be configured as accelerators for neural network applications [65]. However, many other memory technologies like conductive bridging RAM (CBRAM), PCM, filamentary RRAM, non-filamentary RRAM,
magnetic RAM (MRAM), and spin transfer torque MRAM (STTMRAM) is also used to produce artificial synapses based on their desirable properties and switching dynamics as networks of NVMs can perform certain computational tasks with remarkable efficiency [66]–[72]. Although some interesting computational tasks have been performed, the absence of global learning architecture to accompany the local STDP learning rule remains an important limitation in this process. In addition, the inherent random and deterministic imperfections of this kind of NVM devices make more difficulties to achieve the goal. For ANN application, training is a one-time cost, but the prediction phase runs repeatedly once the network is deployed. So, for different machine learning-based applications on IoT devices, weight storage based NN application becomes popular. While memory is being used a weight storage device for IoT enabled machine learning applications, bit density and power play a major role. NAND flash is a highly dense memory and a potential candidate for storing parameters of a NN in a power efficient way.

4.6 Conclusions

In this chapter, we have analyzed the impact of bit error rate of MLC NAND flash memory on the stored model weights and hence the prediction accuracy of neural network. We find that prediction accuracy significantly degrades when BER exceeds 1%. Our analysis shows that the accuracy degradation with NAND flash reliability effects can be explained with random bit error injection with a uniform probability distribution. Using the NAND page-to-page variability, we propose a page priority based new weight storage method which can be used in a more power efficient way in IoT enabled machine learning applications. The proposed reliability-aware page priority based method offers significant advantage over the traditional weight storage method. Experimental evaluation shows that
prediction accuracy remains above 90% with the proposed method even with BER exceeding 1%. We also find that the neural network can absorb the radiation induced BER without losing significant accuracy depending on TID value and the exact type of NAND storage. Recognition accuracy with 3-D NAND weights remains quite robust (>97.5%) with TID ~ 10k rad without any ECC. The ANN accuracy can further be improved if the weight storage algorithm manages the NAND variability more intelligently.
Chapter 5: Wear-leveling Technique for NAND Flash to Manage Big Block Issues

The block size of NAND flash memory chips has increased significantly over the years causing “big-block” management issue for the storage controller. The problem is getting worse with the introduction of 3-D NAND which shows significant layer-to-layer process variation. In this chapter, we demonstrate that the bottom and top layers of a 3-D NAND memory block have significantly lower endurance compared to the middle layers. We find that erase speed variation between the layers is the root cause of the observed endurance variation. Additionally, we propose a new layer-by-layer wear-leveling technique to enhance the lifetime of 3-D NAND memory.

5.1 Introduction

Non-volatile flash memories, the basic building blocks of solid-state storage devices, offer small form factors, high-capacity, high-speed, and low-power permanent storage solutions. They are used for storing and managing increasingly large quantities of data produced every day by over 4.4 billion Internet users and a growing number of Internet-of-Things (IoT) devices. In addition, flash memory is commonly used in a wide range of computing systems found in consumer electronics, automotive, military, industrial, healthcare, and enterprise segments.

The bit density of NAND flash memory has increased by orders of magnitude over the last decade. The introduction of the 3-D NAND technology promises to continue the onward march of bit density growth in the future as well. The new dimension for the 3-D
NAND flash memory is the number of vertical layers which directly increases the bit density for a given die area. Thus, the new “scaling” trend for the 3-D NAND technology is the incorporation of an increased number of memory layers from one generation to the other. However, the increased number of memory layers will introduce several new challenges for the NAND controller that was not present in the case of 2-D NAND. For example, the block size in terms of the number of pages will increase significantly for the future generation of 3-D NAND memory compared to the state of the art 2-D NAND chip. Since each layer of 3-D NAND structure constitutes one or more pages to a corresponding block, the number of pages per block is bound to increase with the increased number of layers in the 3-D stack in the coming years (see Figure 5.1(a)). Managing the large-sized blocks in the memory array is a significant challenge for the storage system controller as it leads to higher write amplification during the garbage collection process.

The second problem with the future generation of 3-D NAND memory is the severe layer-to-layer reliability variation due to the unique nature of 3-D array architecture and the underlying fabrication process. The 3-D NAND fabrication relies on the Reactive Ion
Etching (RIE) process which introduces cell dimension variability across different layers of the 3-D stack. Thus, within the same memory block, the pages located at different memory layers will have significantly different reliability and endurance characteristics. In Figure 5.1(b) we show the relative endurance characteristics of different memory pages within the same memory block of 3-D NAND memory chip. For clear illustration, we divide the memory pages within the 3-D stack into three different groups: bottom layer pages, middle layer pages and top layer pages. We find that the memory pages at the bottom layers/top layers of 3-D stack degrade at a much faster rate compared to the middle layers. Since standard NAND memory controller typically retires a block based on the endurance characteristics of its worst-case page, the endurance variation of memory pages within the same block will lead to severe underutilization of 3-D memory. Thus, 3-D NAND needs to implement a new wear-leveling technique for a given memory block in order to ensure proper utilization of NAND memory.

In this chapter, we find the endurance effect on a flash memory due to process variation and how this affects to the utilization of a chip. Based on our observation we propose a new method for wear-leveling so that the utilization can be maximized. In short, the following are the key contributions of this chapter:

1) We propose a new layer-to-layer wear-leveling technique to enhance the utilization, which can be implemented by any Flash controller without any hardware modification.

2) We experimentally demonstrate the effect of erase operation that affects differently for different endurance, and this is the root cause that is responsible for retiring a block earlier rather it is supposed to be.
3) We explain the tapered based structure is the reason for uneven erase operation for which different layers cell voltage distribution shows significant inequity although they are in the same block.

4) We perform several reliability tests such as high-temperature baking, cycling, etc. and find that the top and bottom layers are more error-prone than the middle layers. Although, all the layers lie in the same block, due to the NAND flash stack-based structure, top and bottom layers become more vulnerable.

5) In general, our proposed method enhances the chance of utilization than the traditional method.

5.2 Background

Layer-to-layer process variation of 3-D NAND: The fabrication process of 3-D memory array is fundamentally different than the 2-D counterparts. Unlike 2-D NAND which poses a planner architecture, the 3-D NAND has a layered structure of alternate

Figure 5.2 (a) 3D NAND flash stack based layered structure where the bottom layers have more diameter than the top layers which makes it tapered. (b) The architecture and file system overview of a flash-based storage system.
metal and insulating layers. Each metal layer forms word line of the memory array connecting multiple memory cells. The channel of the memory cell is formed by creating a hole through the 3-D stack with reactive ion etching (RIE) process. The gate stack of the cells is then formed by sequentially depositing blocking oxide (Oxide-nitride-oxide), charge trap layer or floating gate, and tunnel oxide along the sidewall of the cylindrical trench. Finally, a thin layer of poly-Si channel is deposited with hollow core filled with oxide. The cell structure of 3-D NAND, thus, resembles vertical-channel gate-all-around (GAA) MOSFET. The tapered shape of the cylindrical memory hole (as shown in Figure 5.2(a)) results from the high aspect ratio making the RIE inefficient at the bottom layers. Thus, memory cells at the bottom layers typically have lower diameter than the top layers.

Flash Translation Layer: In order to efficiently manage the flash array’s special characteristics, a firmware layer called Flash Translation Layer (FTL) is typically used by the storage system which is discussed in section 3.2.2. It bridges the gap between the host file system and physical NAND flash memory chips, providing an interface to the host file system on one side and an interface to physical NAND flash memory chips on the other side. FTL provides a block access interface to the host file system by mapping the logical addresses in a block layer to physical addresses in NAND flash (Figure 5.2(b)). In addition, FTL contains firmware modules that perform garbage collection and wear-leveling [73]. The garbage collection module periodically reclaims all the invalid pages in the media to perform a block erase operation, which will free up memory space for new data. The wear-leveling module manages the limited endurance of the flash media by ensuring uniform program-erase operations across different NAND flash memory chips and blocks within a chip.
5.3 Motivation

This section describes the fundamental nature of the problem of underutilization of NAND flash memory blocks in the presence of large layer-to-layer endurance variation.

5.3.1 Underutilization of 3-D block due to layer-to-layer endurance variation:

Big block issue: Block size of NAND flash memory has increased significantly over the years due to technology miniaturization. With the introduction of 3-D NAND, the block size increase will accelerate even further. Since each layer of 3-D NAND structure constitutes one or more pages to a corresponding block, the number of pages per block is bound to increase with the increased number of layers in the 3-D stack in the coming years. Managing the large-sized blocks in the memory array is a significant challenge for the storage system controller as it leads to higher write amplification during the garbage collection process. More importantly, large-sized blocks may lead to underutilization of the memory blocks in the presence of endurance variation between the pages, which is a significant concern for 3-D NAND array as explained in the following paragraphs.

Layer-to-layer Endurance Variation: We find that the different layers of the 3-D stack within the same physical block have strikingly different reliability characteristics, which has been recently reported by other researchers as well. Figure 5.3 and Figure 5.4 summarizes our reliability characterization results on different layers of 3-D NAND memory. Figure 5.4(b) illustrates the different physical layers and the corresponding page distribution for a given sub-block of the Micron MLC memory chip that we used for the reliability characterization. A complete memory block consists of multiple identical sub-block structures (the Micron chip has 16 sub-blocks). Each layer within a given sub-block
contains two shared pages which are called LSB and MSB page respectively. The cell $V_t$ distribution of the MLC memory has four different levels. First, LSB page is written and then MSB page is written which involves cell programming to the highest $V_t$ state. For the reliability characterization, we program the entire memory block with random data pattern, and we read back the data immediately after programming. The raw BER on different memory pages are shown in Figure 5.3. As expected, we find very low BER on the fresh memory block and there is no significant layer dependence. However, on PE cycled memory blocks we find that the BER is significantly increased (immediately after

Figure 5.3 PE cycling and data retention effect on 3D NAND flash in terms of BER vs layer number: (a) For a fresh block without any data retention effect, BER is very low and it does not show layer dependency (b) A 1000 PE cycled block shows layers dependent BER. (c) Fresh block with 150°C HTDR shows little layer dependency on BER. (d). A 1000 PE cycled block with150°C HTDR shows huge variation of BER in different layers.
programming) especially on the MSB pages. In addition, we find that after PE cycling effect, BER shows a strong layer dependence, where mid-layer BER is found much lower compared to the upper and lower layers. This result confirms that PE cycling causes uneven endurance degradation on different layers of 3-D NAND.

The BER immediately after programming may not represent the actual endurance status as flash memory is expected to retain data for a longer duration. For more realistic endurance characterization, we perform an accelerated data retention test by baking the memory chip at high temperature (120 °C) for 1 hour. In Figure 5.3, we plot the high temperature data retention (HTDR) test results on different memory layers. We find that BER is significantly increased on the PE cycled block compared to the fresh block. More importantly, we find that the BER variation across different layers is significantly augmented after the HTDR test. Especially, for the PE cycled block, the BER on the lower/upper layer pages is significantly higher compared to the other pages in the same block.

5.3.2 Root-cause analysis:

The layer-to-layer endurance variation is a fundamental characteristic of 3-D NAND architecture. We identify two main reasons behind such endurance variation. The first reason is the layer-to-layer process variation as discussed in section 2. The process variation results in wider variation of cell diameters in the bottom layers/top layers causing slightly higher BER on the memory pages at the bottom and top of 3-D stack. This effect is widely known and well-documented in the literature. However, the second reason is related to the unique architecture of 3-D NAND memory array which fundamentally dictates uneven erase speed between different layers of the 3-D stack. We illustrate this
uneven erase speed across different memory layers in Figure 5.4(a), where we plot the percentage of non-erased bits (y-axis) on different memory layers (x-axis) after partial erase operation on a memory block written with all zero data. The different colors in the plot represent different partial erase time. For a given partial erase time, we observe a clear trend for non-erased bits vs layer number. Please note that the percentage of non-erased bits represents the erase speed of the layer, where a higher number of non-erased bits implies slower erase speed. Thus, the partial erase data on 3-D memory block illustrates
that the middle layers are slow to erase compared to the lower- and upper-layer cells. This erase speed variation between different layers has a profound implication on the endurance variation between different layers of the 3-D stack as explained in the next paragraph.

Since the lower layer cells are faster to erase compared to the mid-layer cells, the cells in the lower layer experience a much deeper erase (more negative threshold voltage) condition when all the layers get erased. The layer dependent erase state threshold voltage is illustrated in Figure 5.4(d), where we explain the representative threshold voltage distribution for different layers after the erase operation is complete. Please note that endurance degradation depends on the voltage difference between erased state $V_t$ and the programed state $V_t$. Thus, deeper erase will cause faster degradation on the lower- and upper-layer cells given the fact that programed state $V_t$ remains the same for all the layers.

The layer dependent erase speed difference is a fundamental characteristic given the unique architecture of 3-D NAND array. Since the vertical memory channel is connected to the substrate only through the bottom layer, the erase voltage that is applied through the substrate is more effective on the lower layer cells compared to the mid or upper-layer cells (see Figure 5.4(b)). In some of the 3-D architectures, erase voltage is applied through the bit lines making erase more efficient on the top layers as well. However, the middle layers’ cells remain hard to erase as they are far from the substrate as well as the bit lines. This issue of uneven erase speed between the layers is going to be augmented as more layers are stacked in the future generation of 3-D NAND. The uneven erase speed between different memory layers amplify the layer-to-layer endurance difference as the memory block is PE cycled several times (see Fig. 5.4(c)).
In this section, we discuss our proposed layer-to-layer wear-leveling technique in order to fully utilize the memory block without encountering any uncorrectable ECC failure. The key idea behind the proposal is to skip programming on certain MSB-pages located in the worst-case endurance layers. The program skipping will be performed after a pre-defined number of PE cycle count. The reason for skipping programming on MSB pages is to reduce the degradation rate on that layer so that the other pages (such as LSB page in MLC block) remain safe. Figure 5.5 demonstrates the proposed algorithm, where we show the LSB and MSB page distribution across different layers. Since our pre-characterization shows higher endurance degradation for the bottom and top layers‘ cell, we propose to skip programming at the bottom and top layer pages at the beginning. With a higher number of PE cycle count we extend the program skipping steps on the adjacent layers from the top as well as bottom as shown in Figure 5.5. The exact number of PE cycle count \( (n_{PE1}, n_{PE2}) \) at which the program skipping process will be triggered depends on the pre-characterization results of that particular class of memory chip. The proposed design
inherently involves a trade-off between memory capacity and memory reliability. Since the design involves skipping a few pages, it will decrease the effective memory capacity over time. In return, it will significantly improve the worst case BER for a given memory block and hence it will extend the lifetime of the memory block.

5.5 Experimental evaluation

Experimental Set-up: For this evaluation, we have used NAND chips, with multi-level-cell (MLC or 2-bit per cell) storage, from Micron Technology. The part number for the NAND chip is MT29F256G08CBCBBWP-10: B (32-layer 3-D NAND). The rest of the set-up procedure is the same as discussed in section 4.3.1.

Lifetime Enhancement vs. Capacity Loss: Figure 5.6 summarizes our endurance evaluation for a given block in terms of worst-case BER and PE cycle count along with HTDR effect. For a clear explanation, we divide all the pages of a memory block into three groups: bottom layer pages (layer 3 to layer 12) middle layer pages (layer 13 to layer 22) and top layer pages (layer 23 to layer 32). The first and last two pages are not shared, they
are single-level cell (SLC) mode pages and shows higher resistance to BER, so we did not consider those pages for this vulnerability issue. We plot the maximum BER from the pages from each group as a function of PE cycle count. In order to illustrate the data retention effects, we find this worst-case BER obtained after high-temperature bake (1 hour at 150°C). The result shows that cycling with data retention effect affects the top and bottom layers significantly than the middle layers. As a result, with higher BER in a layer makes a block becomes unusable and that block becomes wear-out quickly although lots of other pages still remain usable on the same block. Figure 5.7 also explains that with the shifted read options, although the BER reduced significantly but the layer dependency of BER is still there.

5.6 Conclusions

In this chapter, we propose a layer-to-layer wear leveling technique to extend the lifetime of flash memory. As NAND flash is trending towards higher storage capacity, wear leveling becomes one of the major issues for this device. Efficient wear leveling is
becoming challenging as the block size increases in terms of pages. A block becomes wear out when one weak page of the block shows high BER which can’t be resolved by ECC. We observe that 3D NAND flash shows a large process variation and its bit error rate varies with its vertically stacked layer because of its uneven distribution of cell voltage due to inequitable erase operation. We show that over the course of time and usage of the block, BER of top and bottom layers significantly greater than the middle layers. Because of this, a block is abandoned due to some bad layers despite having some usable layers to the block. As a result, the flash device wears-out quickly. To mitigate this problem and extend the lifetime and usage of a flash device, it is important to consider layer to layer wear leveling so that flash can be used to its maximum breadth.
Chapter 6: Conclusions and Future Research Directions

In this thesis, we explore the security vulnerabilities of NAND flash memory and its countermeasure. At the same time, we present the computing reliabilities of NAND flash memory and big block management for high-density flash. Our proposed methods and observations (1) can recover data partially or fully from a scrubbed NAND flash memory, (2) can efficiently scrub the page of a flash memory so that data cannot be recovered, (3) stores most critical bits of a neural network to the most reliable pages of a flash memory to ensure better accuracy for computing, (4) explains the quick wear-out of a flash memory due to high bit error in layer-based structure and finds that uneven erase operation due to process variation is responsible for that and suggests an idea to efficiently manage those pages to extend the lifetime of a flash memory.

First, we observe that scrubbing based erase creates different threshold voltage to the stored data. Because of this, although the general read shows all the data is in programmed state but our partial erasure-based data retrieval technique can recover the original data partially from the scrubbed memory. To erase the data more efficiently, we propose a partial program-based scrubbing technique to make the threshold distribution almost uniform.

Second, we explore the potentiality of a NAND flash memory as a weight storage device for different edge devices. In edge computing, where power is a major constraint, we experimentally perform different reliability issues on NAND flash where NN weight is stored. We find that the MSB is the most critical bits of NN weight and some of the pages
of flash memory are the most reliable pages whereas some pages get affected too quickly. So, we propose to store the MSB bits to the most reliable pages to enhance the computing performance of NAND flash memory when the ECC is compromised.

Third, we find that the 3D memory layer-based structure offers high bit density but at the same time due to its process variation, the top and bottom layers pages wear-out quickly because of its incoherent erase operation. Which in turn reduce the lifetime of a NAND flash memory block and costs the memory space. To address this issue, we suggest an idea of managing the big block which is expected to enhance the lifetime as well as utilization.

This thesis has shown few unprecedented issues on NAND flash memory and we believe there is a lot more to do in this field. In chapter 3, we have shown that the erase is a costly operation and page-level data erasing is a big issue for flash memory. So, it is a good window to explore more and efficient scrubbing of NAND flash will be a great help for this community. Also, in-memory computing is taking place quickly and flash almost has all the features to be used as a computing memory device. As 3D NAND flash is taking place, managing big blocks efficiently is always an issue and more work can be done in this direction.
REFERENCES


Appendix I

List of Publication(s) from this thesis:


