EXPLORING MANAGED NAND MEDIA ENDURANCE

by

Mark G. Jurenka

A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in Computer Engineering

Boise State University

May 2010

BOISE STATE UNIVERSITY GRADUATE COLLEGE

DEFENSE COMMITTEE AND FINAL READING APPROVALS

of the thesis submitted by

Mark G. Jurenka

Thesis Title: Exploring Managed NAND Endurance

Date of Final Oral Examination: 06 April 2010

The following individuals read and discussed the thesis submitted by student Mark G. Jurenka, and they also evaluated his presentation and response to questions during the final oral examination. They found that the student passed the final oral examination, and that the thesis was satisfactory for a masters degree and ready for any final modifications that they explicitly required.

Sin Ming Loo, Ph.D.	Chair, Supervisory Committee
Jacob Baker, Ph.D., P.E.	Member, Supervisory Committee
Nader Rafla, Ph.D., P.E.	Member, Supervisory Committee

The final reading approval of the thesis was granted by Sin Ming Loo, Ph.D., chair of the Supervisory Committee. The thesis was approved for the Graduate College by John R. Pelton, Ph.D., Dean of the Graduate College.

ACKNOWLEDGMENTS

I would like to take this opportunity to express my appreciation to Dr. Sing Ming Loo. His guidance, supervision and source of education have been instrumental throughout my study and research at Boise State University. I also wish to express my appreciation to the other committee members, Dr. Jacob Baker and Dr. Nader Rafla, for their help and support.

I wish to extend my gratitude to Micron Technology, specifically Brent Lindsay, for the opportunity to pursue and finance this goal. I would like to also express thanks to Jacob Brinkerhoff for his technical assistance during my research.

I would like to dedicate this thesis to my mother, Ethel, for her constant love and support. She has been a source of encouragement and inspiration during this effort.

ABSTRACT

Flash memory can be found in media players, cameras, cell phones and portable storage. These consumer items have universally compatible storage devices. However, what is their longevity and what is the long-term data retention reliability? This thesis will explore and attempt to answer these questions. Predicting accurate endurance ratings and long-term storage reliability is problematic; a storage card in a cell phone will simply wear differently if used for personal computer backup. Advertised longevity ratings can also be ambiguous, specified in a number of years of 'typical' and 'average' use.

This thesis begins by exploring the operation of flash technology used in managed NAND devices. Operational and hidden byproducts of controlling flash memory were identified then directly observed on a sampled MultiMediaCard (MMC) card. The collected data was graphed to calculate the life span of the product for several synthetic data transfer categories. Combined with the total storage capacity, the factors used in longevity calculations are shown to be dependent upon the transfer method.

To answer the original question, a hypothetical camera file storage usage model was contrasted against measured wear data to calculate longevity. When changing the addressing randomness of writing data to fifty percent of total transfers, the 10-year advertised longevity was shown diminished by half. This demonstrated how data storage randomness of the usage model influences device longevity.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

CHAPTER 1 INTRODUCTION

1.1 Introduction

What is the longevity of a portable flash memory storage device? That is the question this thesis will attempt to answer. Portable flash memory storage devices have become popular due to their small form factor and low cost. They are usually based on NAND flash memory technology. Increasing demand driven by product interoperability has been instrumental in the inclusion of flash with an embedded controller (referred to as managed NAND). The result is a standard interface compatible across many different personal computer hardware platforms and portable electronic products (e.g. cell phone, GPS, music players, embedded systems, photography, and personal data storage). At the writing of this thesis, there are various types of NAND flash devices on the market. They include USB thumb drives, MultiMediaCard (MMC), Sony Pico, Memory Stick, Compact Flash (CF), and Secure Digital (SD). They offer consumers a modest storage capacity while maintaining a portable, removable and convenient form factor. However, due to the constraints of the physical interface, non-standardized endurance specifications and low manufacturing cost, these devices satisfy a modest performance requirement.

NAND flash memory used in portable storage devices have operational restrictions and temporary or permanent failures [1]. Because of these issues, external management of NAND flash is required to extend the life of the memory and guarantee

data reliability. However, the algorithms used for management introduce the byproduct of Write Amplification (WA). The definition of WA and methods to measure it vary and accuracy may be influenced by hidden NAND management effects. The equations for end of life, or longevity, use WA and the Long term Data Endurance (LDE) rating or Terabytes Written (TBW) capacity of the flash. The thesis demonstrates how the data transfer usage model significantly affects WA and the longevity of a Managed NAND storage device.

A modified MMC managed NAND device with all internal NAND flash signals bonded to the external package is used. A method was developed to directly measure WA on the NAND flash die components. Data transfers on the MMC interface were performed to create a correlation between the data (usage model) to the effects of WA (NAND wear). The measured WA was used to calculate longevity and demonstrate how storage randomness directly influences the life span of the MMC device. Without this direct observation or a method to retrieve WA information, it appears to be nearly impossible to determine the life span and remaining data retention capability of the MMC device. Within this thesis, two techniques are presented on how to validate a manufacturer's claim of endurance without direct observation.

1.2 Contributions of This Thesis

Predicting long-term data storage reliability on portable and removable managed NAND devices is difficult. Without a mechanism to directly measure the wear on these devices, relying on ambiguous longevity ratings may be risky for long-term data storage expectations. The thesis explores the technology and reasons behind the finite life span of these devices. Subjecting a MultiMediaCard to various data transfer models demonstrated

how using the device may affect long term data retention. Using this data and discussing methods to calculate end of life, this thesis exposes how interpretations of predicting a life span can vary significantly. This thesis provides a better understanding and exposing the risk of assuming reliable long term data retention on managed NAND storage devices.

1.3 Thesis Organization

A description of flash cell operation and the source of errors inherent in the technology are described in Chapter 2. The operational restrictions of NAND memory configurations are also presented. Chapter 3 and 4 introduce NAND memory wear leveling and methods to represent the byproduct of write amplification (WA). In Chapter 5, a summary of equations shows how WA is used to calculate longevity, or device life, by determining the total write bytes capacity with a usage model. A MultiMediaCard device and a tester hardware platform developed for WA analysis is presented in Chapter 6. Test flows are defined to directly measure WA using various data transfer methods. The data was graphed and presented in Chapter 7. Using equations presented in Chapter 5 and data in Chapter 7, end of life (of the sampled MMC device) when exposed to a hypothetical file storage usage model is shown in Chapter 8. The conclusion contains the thesis summary and future research.

CHAPTER 2 PRELIMINARIES AND BACKGROUND

This chapter provides background material on flash memory to enhance the understanding of the material presented in this work.

2.1 The Flash Memory Cell

Nonvolatile flash memory cells retain data for extended periods of time without power or an erase and program (E/P) cycle refresh (refer to as data retention [2]). A conventional flash cell consists of a single N-Channel MOSFET transistor with an isolated floating gate in addition to the control gate as shown in Figure 2.1 [3]. Charged electronics are trapped or removed on the floating gate [4]. This isolated gate provides a mechanism to change the threshold value of the MOSFET transistor cell, V_t.

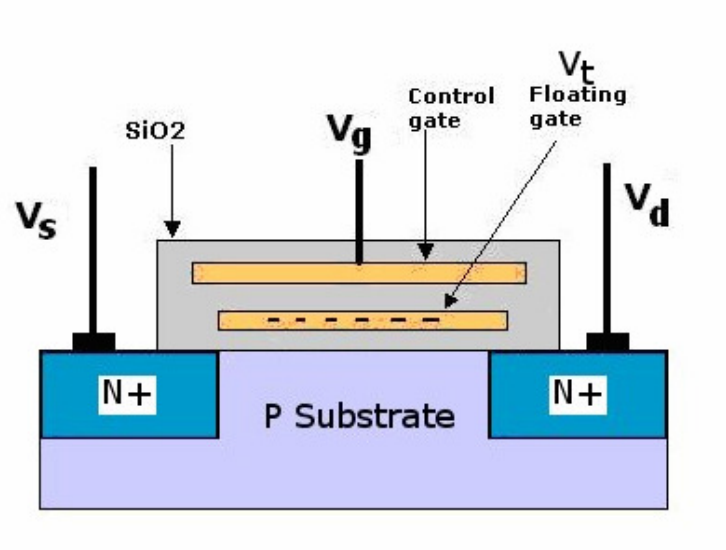

Figure 2.1 Flash Memory Cell

The process called Fowler-Nordheim tunneling moves electrons onto or from the floating gate [5]. The charges on the floating gate directly determine if the cell state is programmed or erased. The dielectric material surrounding the floating gate is degraded by repeated E/P cycles. The resulting dielectric leakage of V_t diminishes the floating gates ability to maintain the programmed charge over time (refer to as wear).

When determining the state of the cell, a reference voltage V_g applied to the control gate is set between the fully programmed and erased V_t of the floating gate. If V_g exceeds V_t, the MOSFET will saturate and is detected as programmed or "0" state. If V_t is greater than V_g, the MOSFET will not conduct to create the erased, or "1", state.

Single Level Cell (SLC) flash memory defines two detectable threshold states. The two programmed V_t thresholds then represent one bit of information. Multiple level cell (MLC) flash memory extends the V_t threshold programming and detection technology to support additional states as illustrated Figure 2.2 [6].

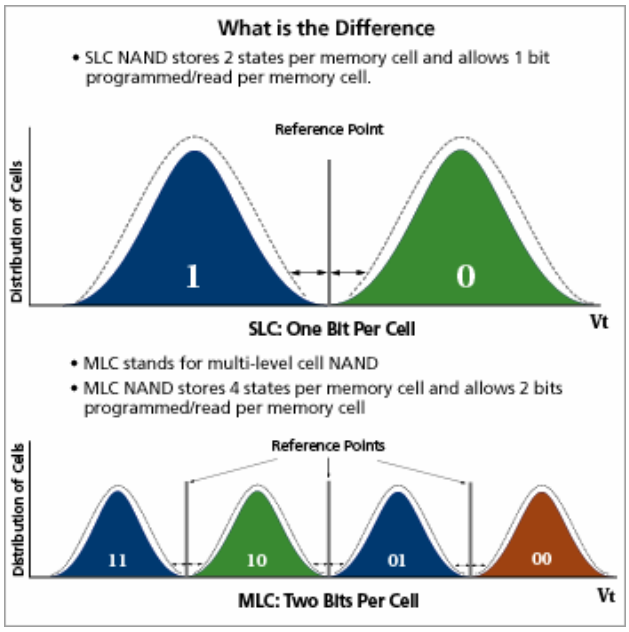

Figure 2.2 SLC and MLC Thresholds

SLC flash offers greater E/P endurance and data reliability. This was the favored technology used in Solid State Drive (SSD) [7]. Fabrication advances in MLC flash technology have increased cell density and have reduced the cost-per-bit. It is the current choice for SSD and consumer portable storage devices. As MLC fabrication geometries shrink to produce greater densities, the endurance, data retention and storage reliability has been suffering [8].

There are two types of flash cell failure: permanent or temporary. Permanent failures are cells stuck in a “0” or “1” state and cannot be refreshed. The defective cell is detected during an E/P cycle and typically retired from use permanently. Temporary failures occur during reading the state of a cell and detecting a stuck or bit-flipped condition. The original state may be logically determined using error correction code (ECC) algorithms and/or physically repaired by an E/P refresh.

2.2 Flash Cell Organization of NOR and NAND

Two common configurations of flash memory cells are NOR and NAND. The advantages and disadvantages of each configuration constrain their target application [9].

2.2.1 NOR Flash

When connected as a two-dimensional array in parallel, NOR flash is created which simulates the logic of a NOR gate. Each memory cell in a NOR device is connected to a common bit line as shown in Figure 2.3. The structure is well suited for random data accessing where each bit can be individually read, erased or programmed. The slower performance of NOR flash sequential accessing is unfavorable for a file data storage device [10].

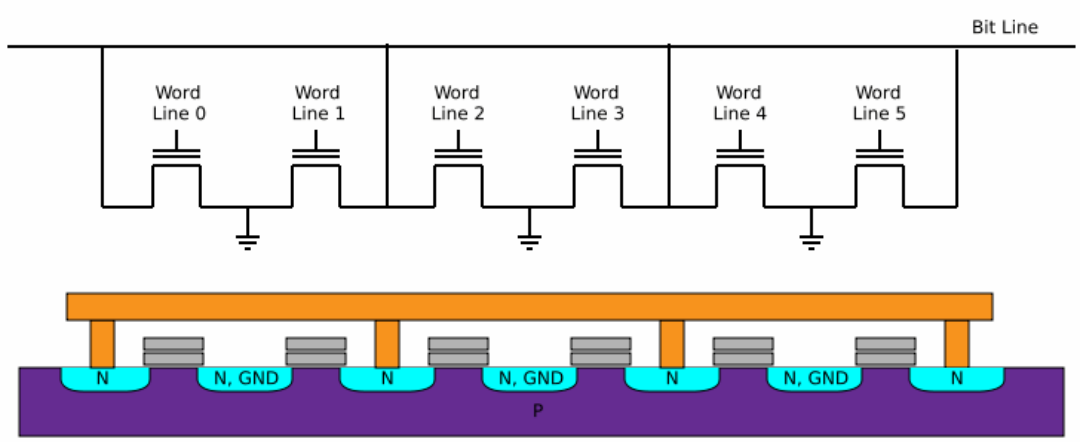

Figure 2.3 NOR Flash Structure

Individual cells of the array are selected by word lines decoded from an external address bus. The resulting cell state is detected on the bit line and applied to an external data bus. There is an inherent accumulating capacitive load effect each cell transistor has on the bit lines. Compared to NAND configurations, a larger transistor, and DIE size, to drive the extra bit-line current is required. The amount of time to write and erase NOR memory is also greater; however an E/P cell endurance rating of 10,000 to over 1,000,000 is typical. NOR flash manufacturers guarantee fault free operation over a specified data retention and endurance rating. This is accomplished by creating redundant row and columns of cells that can be substituted during the fabrication process. NOR applications include a replacement of EPROM or where a processor can execute directly from the nonvolatile memory.

2.2.2 NAND Flash

Flash cells connected in series create a NAND flash configuration which logically operates as a NAND gate. Each array has a single cell transistor connected to a bit line

which minimizes capacitance and transistor die size compared to NOR configurations. NAND flash features greater density, less die cost and faster E/P cycle performance however fabrication tolerances do not guarantee all cells fault free [11]. While efficient for sequential data performance, such as file storage applications, NAND performance suffers during random accesses. NAND cell endurance ratings are typically 10,000 E/P cycles or less.

NAND flash cell arrays are organized into pages, blocks and planes. A page consists of a series of flash cells selected by an address decoded onto word lines as shown in Figure 2.4.

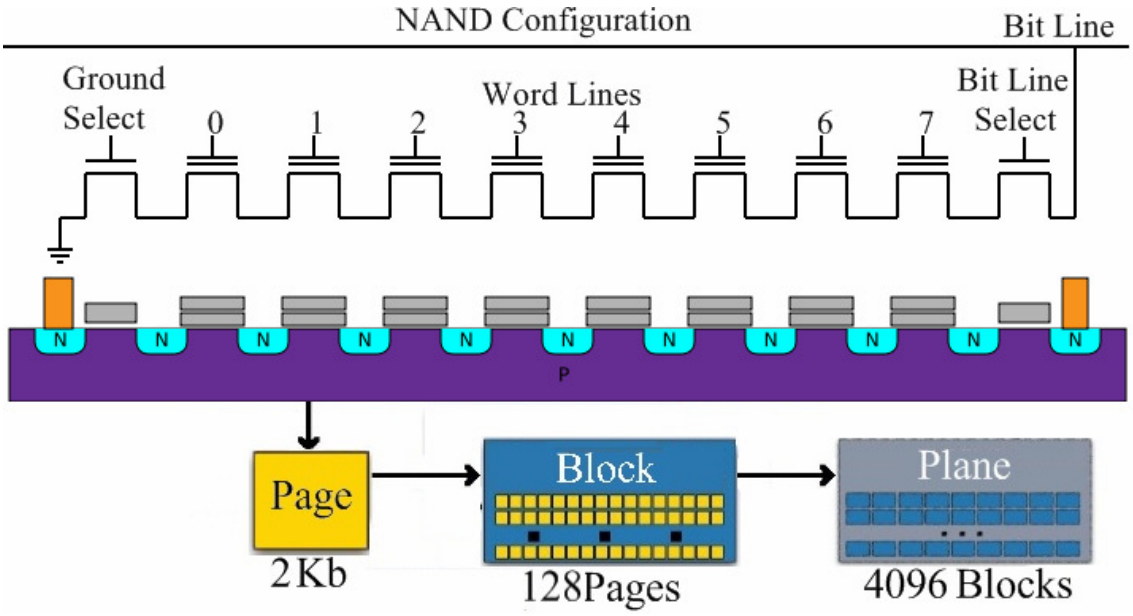

Figure 2.4 NAND Flash Structure

A page of cells includes reserved storage for Error Correction Code (ECC) information. ECC is calculated from the written data and programming into the reserved storage area. It is required by all NAND flash, RAM, hard disks or any other device

susceptible to errors such as random (soft), permanent (hard), temporary (retention and read disturb) and programming (disturb) errors.

Multiple pages are combined together to form a block of data. Programming one or more pages must be preceded by erasing the entire parent block of pages. This is the principle mechanism in causing write amplification (WA) effects. Multiple blocks of flash data create a plane. Multiple planes may exist on each NAND die.

A multiplexed interface of addresses, commands, and data sharing pins under a packet based communication protocol is used. The Open NAND Flash Interface standard (ONFI) attempts to maintain pin and controller consistency over part densities, manufacturers, etc [12].

2.3 NAND Operational Restrictions and Failures

NAND flash memory used in portable storage devices have operational restrictions and temporary or permanent failures. Temporary errors include program disturb, read disturb, over programming, random read bit errors, data loss from diminishing data reliability and retention over time. Permanent errors include bad blocks of data and failures caused by a limited cell erase and programming (E/P) endurance. Because of these issues, external management of NAND flash is required to extend the life of the memory and guarantee data reliability. The algorithms used during management compensated for the following operation restrictions and failures.

The NAND cell configuration and control logic restricts a minimum program size to one sequentially written page of cells within a parent block. Partial page programming is typically not permitted although pages may be left erased and skipped within the block. A page within a block cannot be reprogrammed or individually erased. During

programming, temporary Program Disturb errors occur when an unintended V_t charge collects in an unrelated cell, changing from a '1' state to a '0' state. Programming new data on a previously programmed page, including leaving cells at the erased state of '1', introduce write disturb errors on upper erased pages. Over Programming errors occur when a cell threshold gate voltage on a bit-line within a block is excessive, preventing the cell to be read.

Read disturb errors are caused when repetitive read operations on a page induces a V_t change in one of the other addressed cells within the page. Typically hundreds of thousands or more read operations are required. Random Read bit errors are caused by several mechanisms including V_t threshold interpretation errors from the sense logic on the bit lines. This is primarily due to cell wearing causing leakage to violate initial programming tolerances for correct cell state detection.

Repeated E/P cycles rapidly deteriorate cells by diminishing their ability to reliably maintain their programmed state over time without a refresh (referred as endurance). Hard bit failures occur when the cell failed to program after internally trimmed timeouts conditions are reached. When this failure status is detected, the parent block must be retired after moving any remaining page data to a functioning block.

The Joint Electron Devices Engineering Council (JEDEC) specifies a relationship between endurance and data retention measured in years [13]. It states that 100% of device endurance is the number of E/P cycles considered to be fully worn or guarantee a 1 year data retention capacity. At 10% of device endurance, the minimum number of E/P cycles that still guarantee a 10 year data retention is specified. This standard may not be fully implemented by the Manufacture, offering a 10 year data retention initially until

10% of the rated E/P cycles are reached before accepting a 1 year retention rating. Manufacturers determine the number of E/P cycles a NAND flash can be exposed to and guarantee data retention by taking a sample of devices and wearing under a chosen usage model. By exposing it to high temperature over a determined amount of time to stress the part, the device is read to verify the data and check for ECC failures. This standard is called distributed testing.

CHAPTER 3 MANAGED NAND

A standard NAND flash interface includes control and data bus signals directly communicating to the memory. In addition to the physical signals, communication algorithms using the packet protocol of the NAND flash must be implemented. ECC, wear leveling and bad block management activities are the responsibility of the host processor. Portable consumer storage devices such as MMC use an embedded processor to manage the NAND flash memory. The controller uses proprietary algorithms to wear the cells evenly (to extend longevity), retire failed cells and correct read bit errors. The controller implements a standardized interface protocol to the host regardless of internal NAND configuration, size, and operation. The MMC interface and protocol standard is defined by JEDEC [14]. The external interface may be completely abstracted from all internal management activities. Figure 3.1 shows the differences between a standard NAND flash interface and the MMC managed NAND memory.

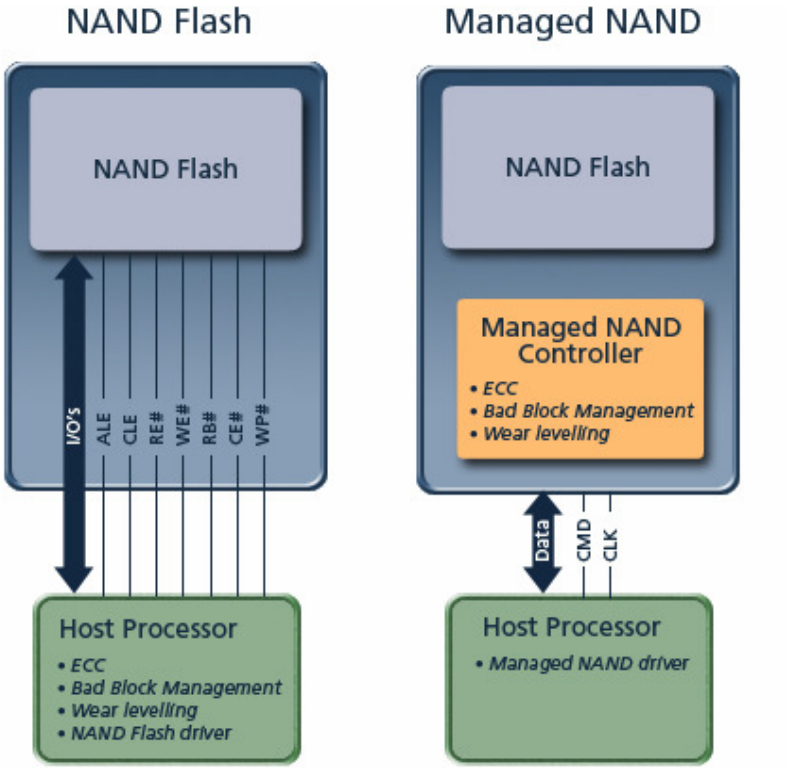

Figure 3.1 NAND Memory Bus VS Managed NAND Interface Bus

The MMC interface protocol was optimized for data transactions consisting of 512 byte sectors, a common size used in hard drives [15]. The storage device is represented as a sequential address range of sectors defined by a logical block address (LBA) table. The host processor application may read or write any LBA sector in any valid order or range; repeatedly, randomly or sequentially. Internally, the data is written and moved over the entire device to evenly distribute E/P cycles on every available and spare area block (referred to as Wear Leveling). Figure 3.2 shows the abstraction between the MMC host interface (LBA Table) and the distributed storage on the internal NAND flash.

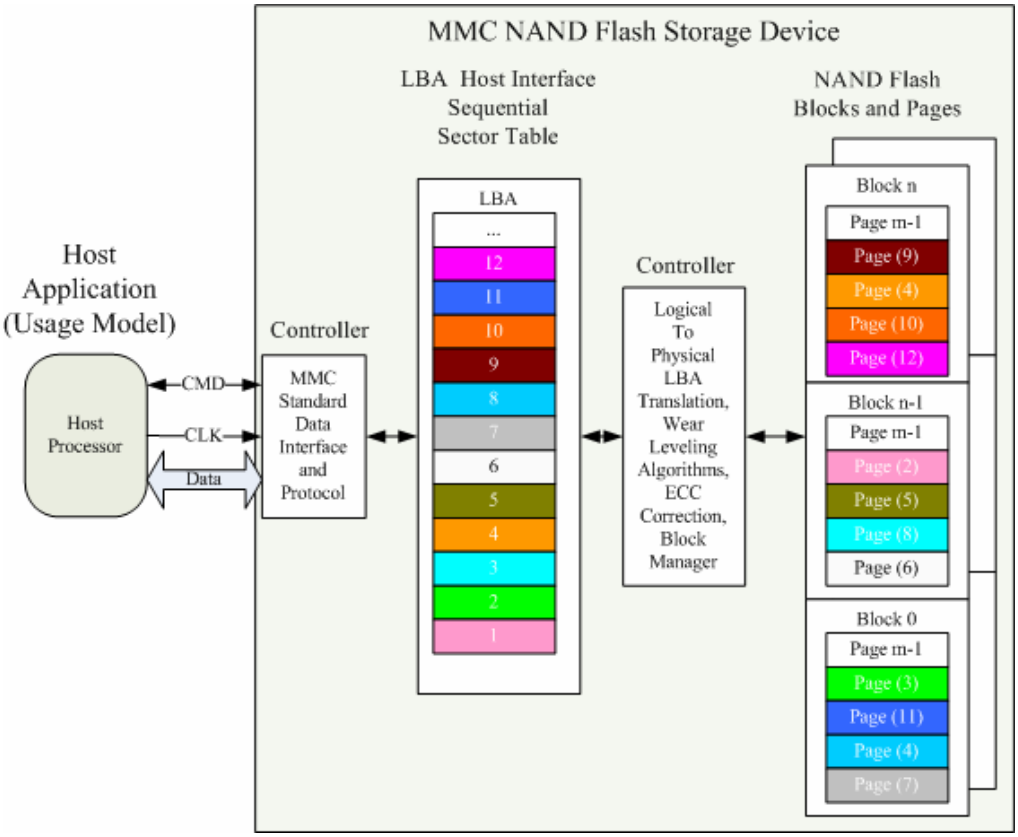

Figure 3.2 MMC Controller LBA Translation

Managed NAND controllers incorporate ECC to correct read-bit failures transparent to the host processor. The ECC information is programmed into the dedicated spare bytes of each page. There are two major factors (error detection ability and error correction ability) in measuring the effectiveness of an ECC algorithm. Hamming (SLC), Reed Solomon (MLC) and Bose-Chaudhuri-Hochquenghem (BCH for MLC) are the most popularly used ECC algorithms [16].

One method to increase endurance (by reducing E/P cycles and improving random data performance) is using cache memory. Fragmented LBA transfers written by the host are collected, organized then written to the NAND memory as illustrated in Figure 3.3

[17]. However, there is little to gain from this architecture because of the modest performance and low cost requirement of flash storage devices.

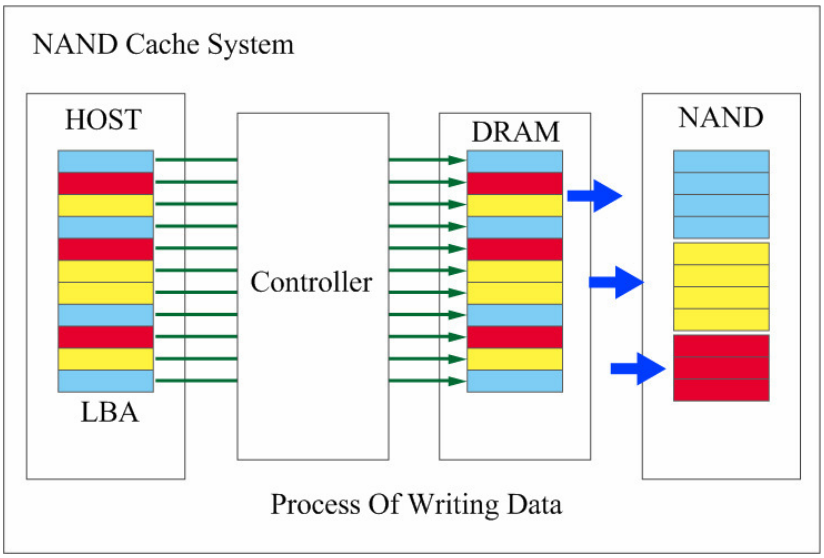

Figure 3.3 Cache Operation

3.1 Wear Leveling and Flash Management

To extend device longevity, writing to NAND flash requires a method to reduce the number of E/P cycles on any one individual block. The controller algorithms erase, program and move data across the NAND flash as needed to evenly wear all blocks of memory. Unfortunately, moving data may trigger additional E/P cycles that increase write amplification effects. As data is moved, temporary errors may occur on previously programmed data. If the ECC algorithm fails to correct the page data during a read, the block may be marked as bad (referred to as ECC retirement). Two common methods of wear-leveling techniques are summarized in Figure 3.4. They consist of Dynamic and Static wear leveling.

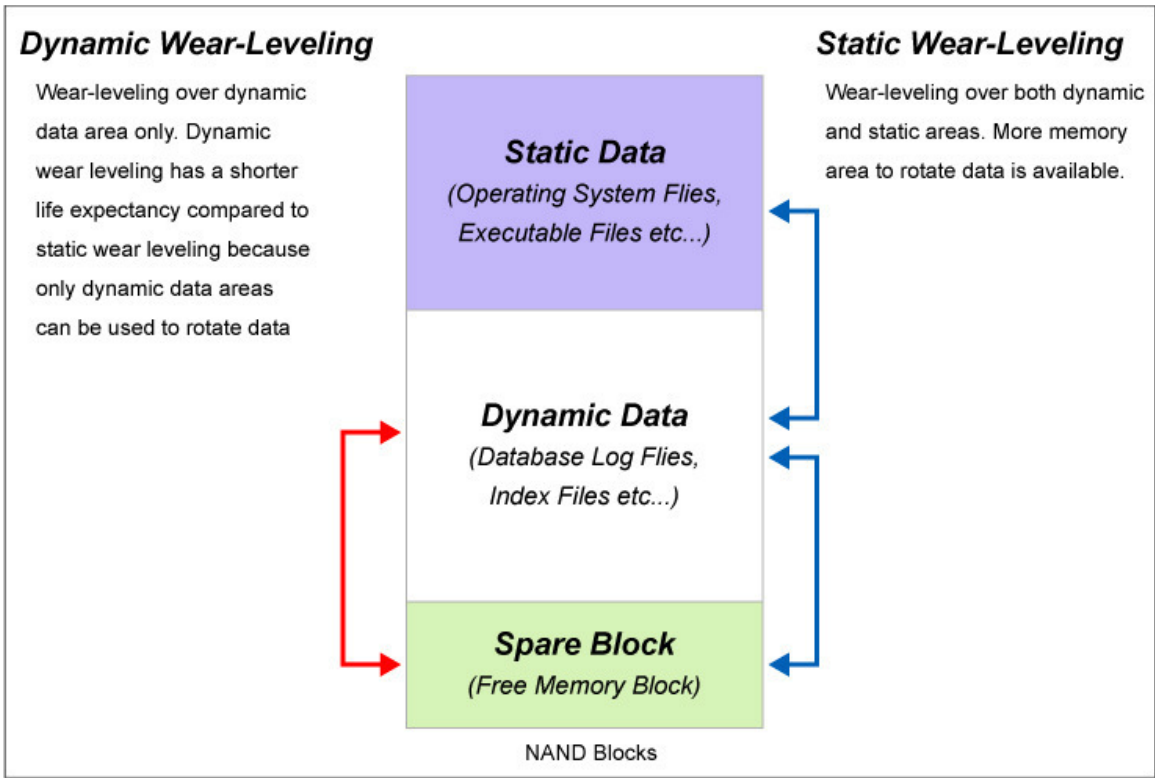

Figure 3.4 Dynamic VS Static Areas

Dynamic wear leveling is the simplest form of block management and is heavily influenced by the host. It consists of rotating E/P cycles through frequently used data blocks (referred to as dynamic area). Rarely or unused blocks of data are left untouched (referred to as static data). The wear on the dynamic area, including spare blocks, increases as more data is written to the static area. As the density of MLC NAND flash technology increases, the cell endurance is decreasing, reducing the number of E/P cycles. Because of this, dynamic wear leveling on MLC flash may not be reliable.

Static wear leveling uses all available data and spare blocks in the device. More complex algorithms move data, erases blocks and programs new data over the entire device in an attempt to wear every block evenly.

All NAND devices have or will have one or more bad blocks marked as unusable. Manufacture yield considerations including process, die cost and testing together permit bad blocks to be randomly scattered over the die. Initial failures of cells are determined though factory testing phases to mark blocks as bad for management algorithms to avoid. As blocks are cycled during operation, they eventually reach their specified E/P rating or fail to program and must be retired. Bad blocks are marked as unusable by the controller, total memory capacity may become reduced over time. The more recent ONFI standard attempts to unify the industry bad block reporting of NAND flash devices.

3.2 Hiding the Issues

The host processor using the MMC device has limited visibility of the quality of the wear leveling, data retention capability, and current or future lifespan of the product. At the writing of this thesis, no standard MMC protocol command or method exists for the host to mark previously written LBA sectors as unused. This creates blind wear leveling activities across all LBA sectors regardless of the data no longer required by the host. MMC commands are available to erase sections of memory but the standard is vague in the definition of recovering used sectors. Wear leveling algorithms begin to move and clean blocks when the NAND reaches, or approaches, full capacity. When full, write performance may decrease along with an increase in WA due to the overhead of wear leveling retired data. Until all sectors are used, the initial WA and write performance measurements may be inaccurate. When initial WA values are used in calculating longevity, the end of life estimations may be inaccurate.

CHAPTER 4 FACTORS MEASURING ENDURANCE AND LONGEVITY

This chapter examines equations for Write Amplification (WA) and Page Program Ratio (PPR). These factors are used to calculate longevity. The Page Erase Ratio (PER) is introduced to represent page utilization

4.1 Write Amplification (WA) Ratio Methods

The value of WA is represented as a constant. Several definitions of WA are presented by manufacturers of NAND storage devices. However, WA does not take into account flash endurance (E/P cycles), over provisioning capacity or the effects of flash operating in SLC or MLC modes.

4.1.1 WA Ratio Using Pages Erased to Pages Written

WA can be represented as the ratio of pages erased on the NAND flash to pages of data written from the host. Each page written by the host is defined as a transfer of data equal in size to one NAND page size. The ratio of the total pages erased to pages written is ideally 1. This indicates no unused pages remain in a block of flash memory after erasing and programming. Equation 4.1 shows this representation of WA.

$$WA = \frac{NANDNumberPagesPerBlock \times BlockErase}{NumberOfPagesWrittenByHost} \tag{4.1}$$

A widely used file storage protocol is the FAT-32 file system. It is commonly used on file systems for PCs, laptops and portable electronic devices. It consists of a data transfer size of 4 KByte clusters. Let's consider a NAND device with a page size of 4 KBytes with 128 pages per block. If the host wrote 128 transfers of 4KByte data, the WA value of 1 (the ideal WA) would indicate a single block erase occurred. This indicates that every page in each block was programmed. Conversely, a WA value of 128 would indicate that on average one page, the minimum was programmed over the erased blocks. Wear leveling activities may create additional E/P cycles to occur, increasing WA

4.1.2 WA Ratio Using Total Bytes Programmed to Bytes Written

Perhaps a more accurate representation of WA may be the ratio of total bytes erased on the NAND to the total bytes written by the host. Equation 4.2 includes the overhead of erasing unused pages in calculating WA. The equation assumes that erasing unused pages wear the cells equally to those programmed. The ideal WA value of 1 indicates that all erased pages were fully programmed.

$$WA = \frac{NumberOfErases \times BytesPerPage \times PagesPerBlock}{TotalBytesWritten} \tag{4.2}$$

4.2 Page Program Ratio (PPR)

Page Program Ratio (PPR) may be used in calculating longevity. PPR represents the ratio of NAND pages programmed to bytes written from the host. Equation 4.3 can be used to calculate PPR. The wear from erasing unused pages is not considered. The ideal PPR value of 1 represents no additional pages were programmed than pages of data written from the host.

$$PPR = \frac{TotalPagesWritten \times PageSizeInBytes}{TotalBytesWrittenByHost} \quad (4.3)$$

4.3 Page Erase Ratio (PER)

Page Erase Ratio (PER) is a ratio of how many pages are programmed on each block between E/P cycles. Using equation 4.4 and considering a NAND with 128 pages per block, the ideal value is 128. This represents all pages within the block were programmed for a block erase. PER may be an indicator of wear leveling algorithm efficiency.

$$PER = \frac{TotalPagesWritten}{TotalNumberOfBlockErases} \quad (4.4)$$

When calculating longevity accurately, a WA value considering all NAND pages (bytes) worn by E/P cycles may be more accurate. As seen in equation 4.2. The calculation uses the total number of NAND bytes erased and bytes transferred from the host. WA is affected by wear leveling and block management activities of managed NAND controllers. The value will also be dependent upon the transfer size and LBA randomness of the written data.

CHAPTER 5 ESTIMATING END OF LIFE

In order to calculate longevity, a Long Term data Endurance (LDE) specification was created [18]. LDE is the amount of data that can be written over the lifespan of the NAND storage device. It is specified in Terabytes Written (TBW). The endurance of the NAND flash was defined as the number of E/P cycles for each block to reach 100% wear. The TBW calculation of equation 5.1 uses the total size of the NAND flash and endurance rating. The spare area and reserved storage used by the controller is included.

$$TBW = NumberOfBlocks \times BytesPerBlock \times Endurance \quad (5.1)$$

An alternative calculation of TBW considers the available storage capacity as reported by the managed NAND controller. The controller divides the available storage area into sectors, each represented by a Logical Block Address (see Chapter 6). Equation 5.2 considers the number of LBA addresses available and sector size to determine TBW.

$$TBW = TotalNumberOfLBA \times SectorSize(bytes) \times Endurance \quad (5.2)$$

Manufacturers may specify their average TBW over best, typical and worst case transfer conditions to include the effects of WA. These include write transfer size, sequential or random distribution and percentage of re-written sectors. Equation 5.3 shows TBW including WA.

$$TBW = \frac{TotalNumberOfLBA \times SectorSize(bytes) \times Endurance}{WA} \tag{5.3}$$

5.1 Equations Using Total Program/Erase Endurance without WA

The life span calculation of equation 5.4 represents a simple relationship between the Terabytes Written (TBW) capacity and the transfer rate of write data. This represents the ideal life (in years) of the device without considering WA.

$$Life(years) = \frac{TBW}{WriteTransferSizeInBytes \times TransfersPerDay \times 365} \tag{5.4}$$

5.2 Equations Using Total Program/Erase Endurance with WA

The life estimate of equation 5.3 was modified to include the WA constant. The life calculated by equation 5.5 reflects the inclusion of flash wear.

$$Life(years) = \frac{TBW}{WriteTransferSizeInBytes \times TransfersPerDay \times 365 \times WA} \tag{5.5}$$

5.3 Equations Using IOPS, Endurance and Drive Capacity

Input-Output-Per-Second (IOPS) is a performance benchmark for storage media [19]. A device life span in years may be estimated using TBW and the write IOPS measured (streaming data). A conversion factor constant generated by equation 5.6 is required to convert IOPS from seconds to years.

$$CF(Seconds) = 3{,}600 \times 24 \times 365 \tag{5.6}$$

Equation 5.7 calculates life using a measured IOPS rating for streaming write transfers. This includes a duty cycle or the percentage of time spent writing to the device. The equation determines the number of years of continuous write transfers before reaching TBW.

$$Life(years) = \frac{TBW \times CF}{WriteSpeedIOPS \times WriteDutyCycle \times WA} \tag{5.7}$$

Life estimation may be represented by the number of file transfers instead of streaming data. The file transfers may be burst or sustained, small or large, random or sequential. Manufacturers have created a software benchmark called IOMETER [20]. By choosing the file transfer size and LBA addressing randomness, this application measures real time write IOPS. Equation 5.8 may be used to calculate life with the measured IOPS value.

$$Life(years) = \frac{TBW \times CF}{WriteIOPS \times FileSizeInBytes \times DutyCycle \times WA} \tag{5.8}$$

5.4 Equations Using PPR

Equation 5.9 uses PPR to measure life in years. The estimation is in terms of data transfers per day. The value of PPR, and resulting life estimate, does not consider the wear caused by erasing unused pages.

$$Life(years) = \frac{TBW}{WriteTransferSizeInBytes \times TransfersPerDay \times 365 \times PPR} \quad (5.9)$$

The equations using TBW assume a constant write transfer model, duty cycle and WA value over the life of the device. The original transfer conditions used to determine WA may differ from those used in the equations. Using a constant WA value may then be inaccurate. Equations using IOPS may use measured write performance at a known (average) file transfer model. However, without a corresponding WA value the equations may not be accurate. The most accurate method to measure life is by measurement and analysis of NAND wear under real time write transfers.

CHAPTER 6 ANALYSIS BACKGROUND AND TESTBENCH

In this chapter, the sampled test device and test platform are described. This includes the hardware, software, firmware, algorithm and test definitions.

6.1 MMC Managed NAND Device

JEDEC MMC V4.4 describes the physical interface, commands and protocol to communicate to the device. The MMC controller divides the total (reported) storage capacity of the device into 512-byte sectors (the minimum transfer size). Each sector is represented by a Logical Block Address (LBA). Write and Read commands specify the starting LBA address for sequential or randomly written sectors. Data transfers can be a single, a predetermined number, or open ended stream of sectors. Open ended transfer modes allow continuous transfers until interrupted by a MMC command. Predetermine and open-ended transfers require the MMC controller to auto-increment the starting LBA. Dynamic and static LBA ranges are determined by the host application.

6.2 DUT Sampled Device Specifications

The Managed NAND device chosen for this analysis is a 4 GB High Capacity (HC) MultiMediaCard with an 8-bit data interface. The MMC interface clock is 30 MHz. The device consists of 4-die MLC flash components, 4,096 blocks per die, 128 pages per block, and 2,048 bytes per page. The NAND flash endurance was given by the manufacturer at 10,000 E/P cycles. This represents 100% wear or the number of E/P

cycles on each block to diminish data retention to one year. Using equation 6.1, the total flash size was calculated at 4,294,967,296 bytes including bad blocks.

$$4Die \times (4096Blocks \times 128Pages \times 2048Bytes) = 4{,}294{,}967{,}296\text{bytes} \quad (6.1)$$

The MMC controller reported a total capacity of the device to be 4,112,515,072 bytes (equation 6.2). The advertised capacity of the device was 4 Gigabytes, a difference of 356,352 bytes.

$$8232256LBA \times 512BytesPerSector = 4{,}112{,}515{,}072bytes \quad (6.2)$$

6.3 Tester Hardware Configuration

The test platform block diagram is shown in Figure 6.1. Due to the internal NAND flash signals of the sampled MMC device bonded to the device package, a custom socket was necessary. The NAND and MMC interface signals of the device were routed to the tester platform. A MMC controller with test flow algorithms was designed in an Altera© FPGA with embedded NIOS II© soft-core processor. The FPGA provided a means to create custom Verilog Hardware Description Language (HDL) modules to interface to the MMC device. The embedded CPU uses the C programming language to write test flows and communicate to a host PC application.

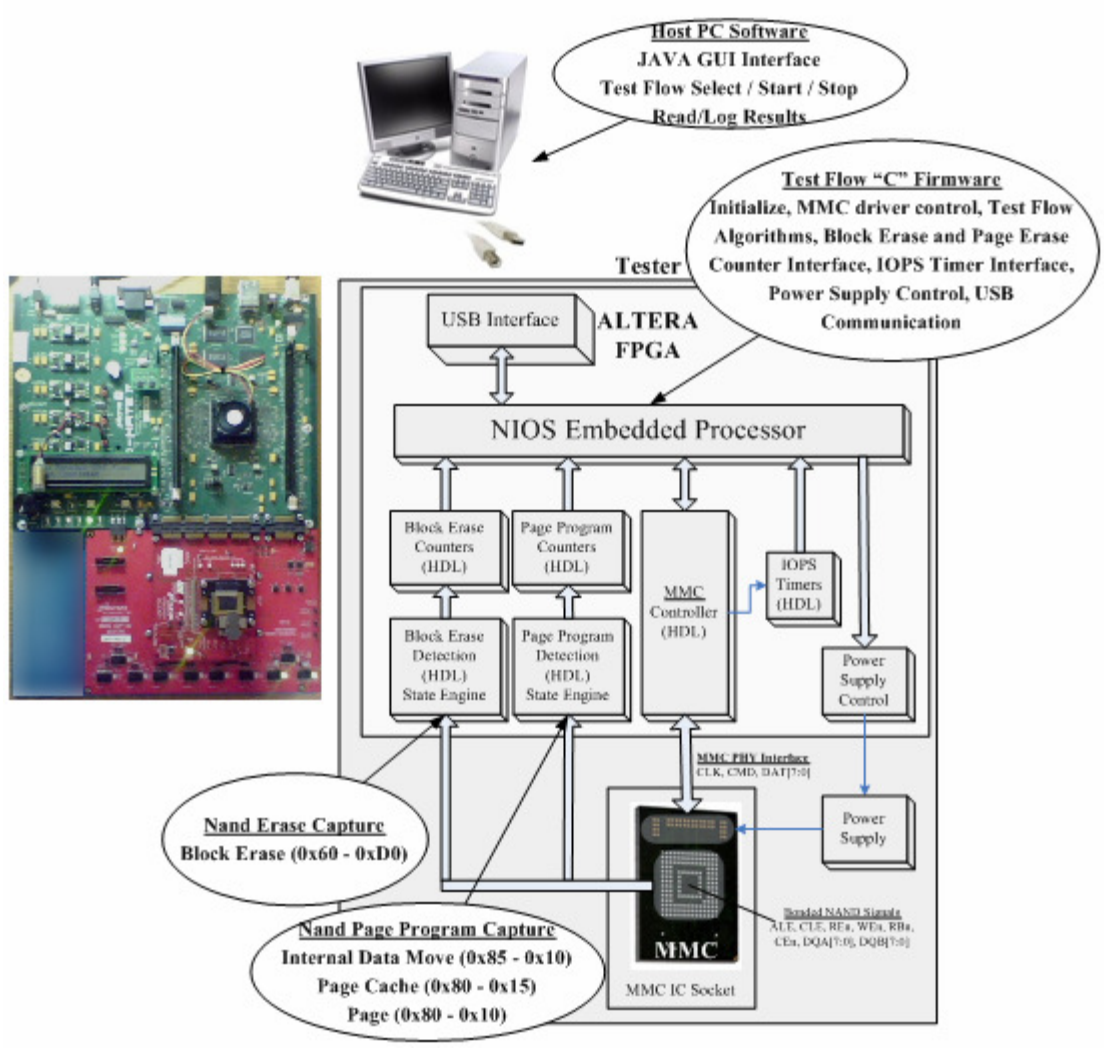

Figure 6.1 Tester Platform Hardware

The erase capture logic detects a NAND block erase command sequence. Upon detection of a completed cycle, the erase event was counted. The page program capture logic detects all NAND page program commands. These include program page, program page cache mode, program for internal data move, and dual plane page program commands. When any of the commands are detected, a page program counter is incremented. The FPGA logic includes timers to measure test duration and write transfer IOPS performance on the MMC interface.

Communication between the FPGA firmware and host PC is accomplished using a USB interface. A Java GUI application on the PC (see Figure 6.3) was written to select the test flow to perform and specify the test parameters. Upon test completion, the erase and page counter values, write IOPS timers and test flow time were retrieved and logged.

6.4 Tester Firmware Algorithm

The test algorithm is shown in Figure 6.2. A test consists of writing clusters of sectors (512 bytes) to the MMC device. The total number of clusters written to complete a transfer is determined by the test flow. Upon reaching the total number of cluster transfers, the test is terminated and results are retrieved. Power was cycled on the MMC device is required at the start of each test flow.

The test flow consists of specifying the size of the write data cluster, number of clusters per transfer, the MMC write mode, starting LBA address and if using sequential or random LBA addressing. The tester calculates the next sequential or random LBA address for each cluster during the transfer. Two MMC write commands were used; open ended and predetermine.

The MMC device was first initialized with an “AA” HEX pattern. This assured all MMC device sectors and spare blocks are written. A random data pattern is used for all test flow transfers.

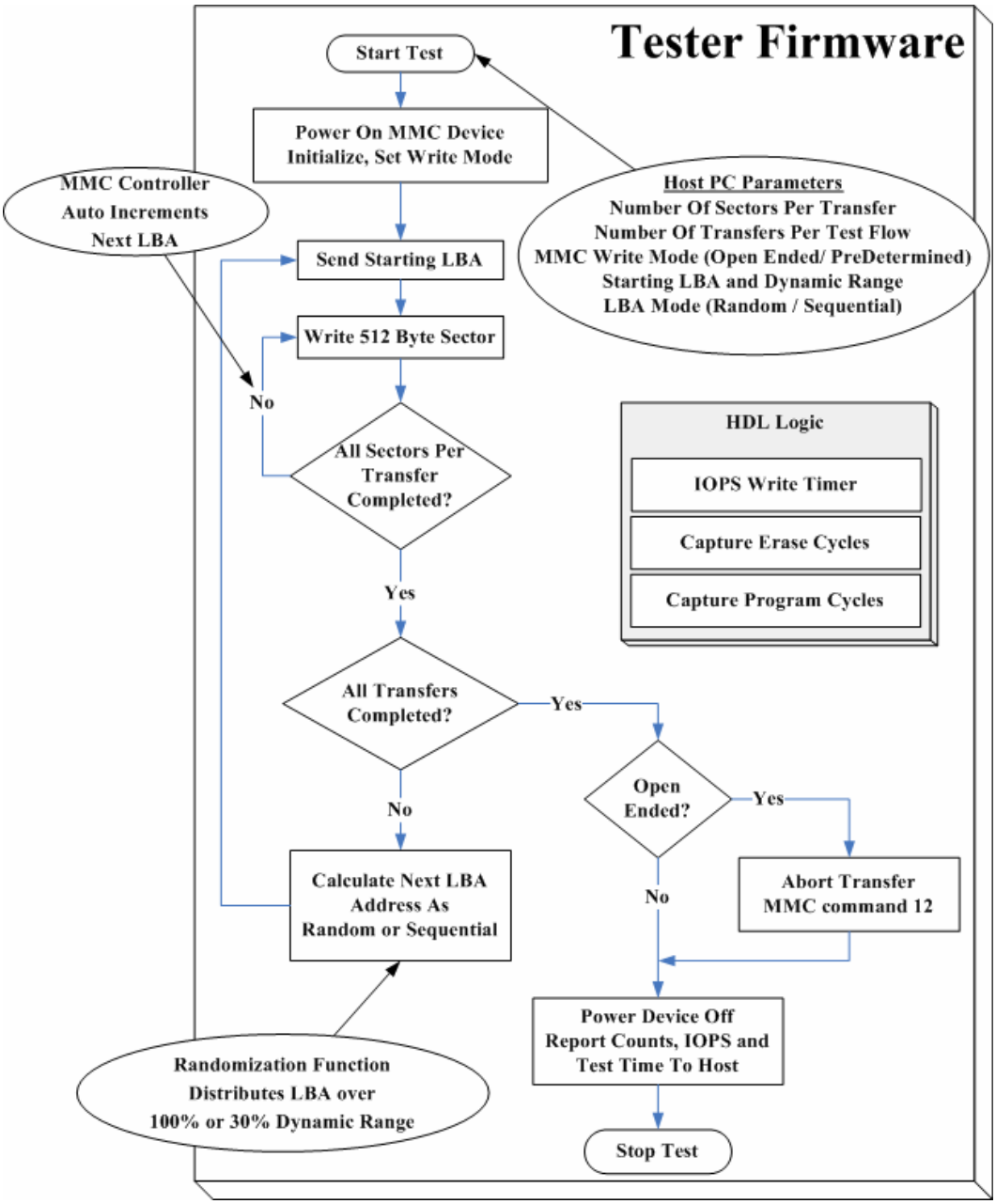

Figure 6.2 Tester Firmware Algorithm

6.5 Test Definitions

Two tests of streaming write transfers were used (Type I and Type II). In Type I streaming write transfer, 100% of the all LBA data sectors are written (4 Gigabytes). For Type II streaming write transfer, the lower 30% of LBA data sectors were written (1.2

Gigabytes). Type II test represents a 30% dynamic area (70% static). Each test required writing clusters of sectors to complete the total transfer size. The cluster sizes of 1, 4, 8, 16, 32, 64 and 128 sectors were chosen. This represented 512, 2048, 4096, 8192, 16384, 32767 and 65536 bytes per cluster, respectively. Upon each test completion, the measured number of erase and page program commands is used to calculate WA, PER and PPR. The results include write IOPS performance and are presented in Chapter 7.

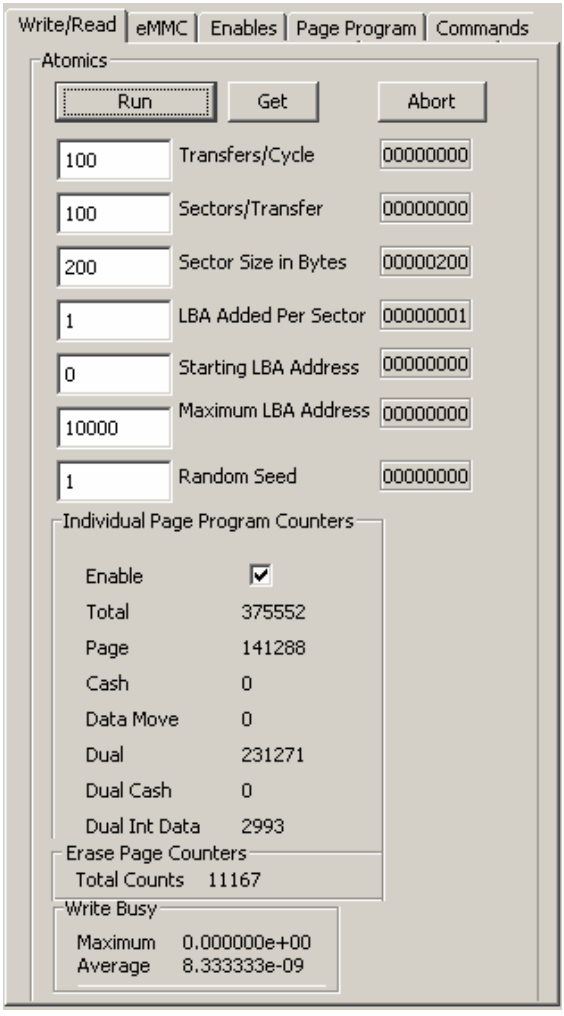

Figure 6.3 Host PC Application (JAVA)

CHAPTER 7 MEASURED DATA AND RESULTS

This chapter determines endurance from test results observed using the hardware test platform and algorithms as presented in chapter 6. The equations presented in Chapters 4 and 5 are used to determine different means of calculating endurance. Each test flow measures WA, PPR and IOPS for each cluster size. Using these values, TBW is calculated. Each graph presented in this chapter requires hours to days to capture the data. For readers interested in the raw data, it is include in the Appendix. For some cases, due to amount of time required to collect the data, we trade-off cluster size against LBA range. For example, since random LBA transfers using small cluster sizes required very long transfer periods, a smaller LBA range was used to collect a representative sample of P/E cycle data.

7.1 Measurement of Write Amplification (WA)

The WA calculation considers the ratio of total bytes erased on the NAND die to total bytes written by the host (equation 4.2). The wear caused by erasing unused pages during wear leveling is included.

7.1.1 Cluster Size VS WA Using Sequential LBA

Figure 7.1 compares WA for Type I and Type II transfers using cluster size and write modes (open ended or predetermined). The calculated WA for each cluster size and type of transfer is shown in Table 7.1. Larger sequential cluster sizes of 4 KBytes or more

appear to exhibit the least WA. The controller appears to perform a significant increase of wear leveling activity for the single sector cluster of 512 bytes (less than the NAND page size). The data for the MMC write modes of open ended and predetermine indicate no significant WA performance difference.

Table 7.1 Cluster Size VS WA Using Sequential LBA

Cluster Size VS WA Using Sequential LBA				
Cluster Size	Open Ended		Predetermine	
	Type 1	Type 2	Type 1	Type 2
512	44.21	44.22	44.21	44.22
2048	8.29	14.27	8.28	14.27
4096	2.33	8.28	2.33	8.28
8192	2.33	5.28	2.33	5.28
16384	2.33	3.79	2.33	3.80
32768	2.33	3.04	2.33	3.06
65536	2.34	2.67	2.34	2.68

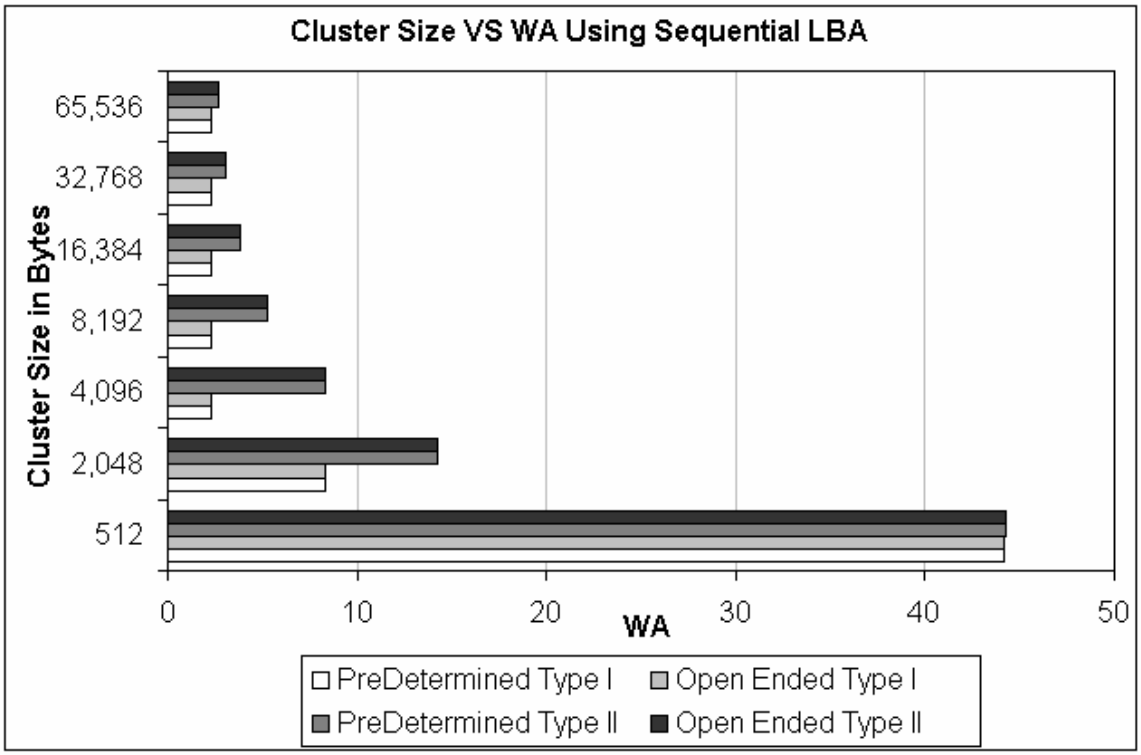

Figure 7.1 Cluster Size VS WA Using Sequential LBA

7.1.2 Cluster Size VS WA Using Random LBA

Type I and Type II transfers using random LBA addresses increased WA compared to sequential addressing (Table 7.1). Using random LBA addresses with a cluster size of 4 KBytes, the WA increased from 2.33 to 2049. The MMC write modes of open ended and predetermine show a small WA performance difference (Figure 7.2). The results shown in Table 7.2 suggest that random LBA transfers with a cluster size smaller than the NAND page size are problematic, creating a high WA value.

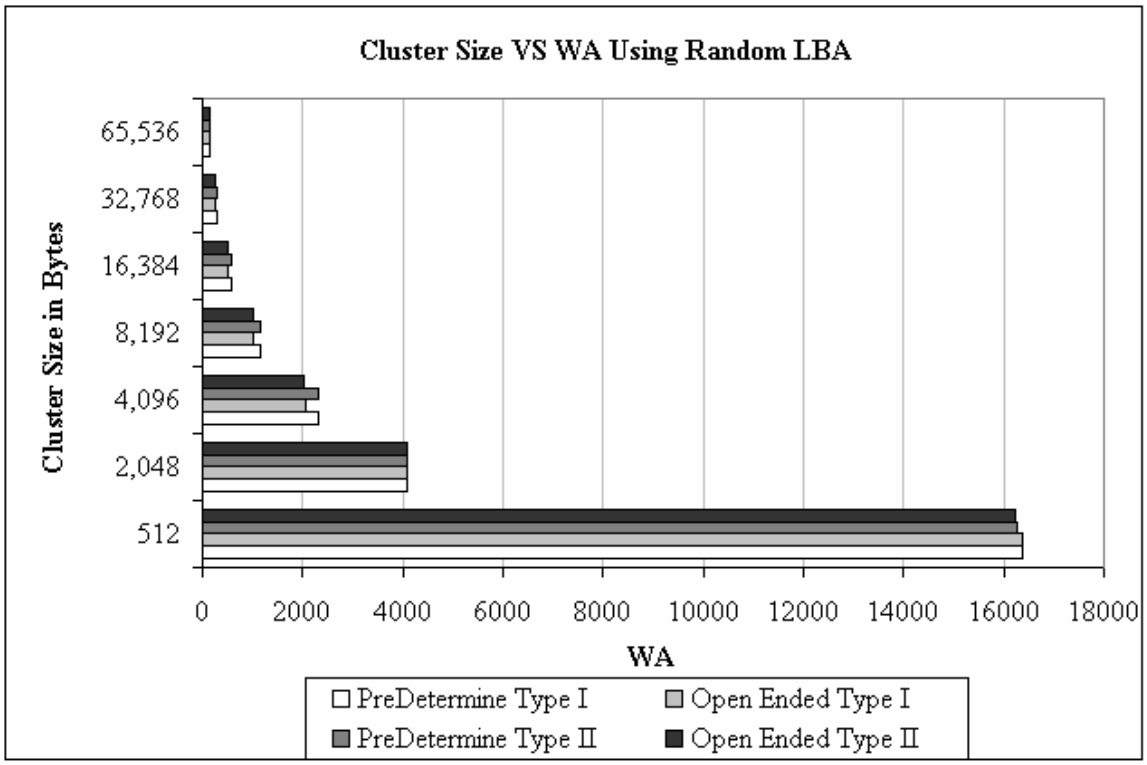

Figure 7.2 Cluster Size VS WA Using Random LBA

Table 7.2 Cluster Size VS WA Using Random LBA

Cluster Size VS WA Using Random LBA				
Cluster Size	Open Ended		Predetermine	
	Type 1	Type 2	Type 1	Type 2
512	16374.99	16240.23	16388.40	16259.48
2048	4098.71	4075.52	4094.31	4075.11
4096	2049.02	2038.58	2307.06	2305.73
8192	1024.99	1024.19	1153.24	1154.57
16384	512.79	514.362	576.55	579.41
32768	256.94	259.74	288.2	292.02
65536	128.94	132.16	144.06	147.73

7.2 Measurement of Page Program Ratio (PPR)

Page Program Ratio may be used to calculate longevity. It considers the number of pages programmed on the NAND to the number of bytes written by the host. However, the overhead of erasing unused pages is not considered. The results are compared to the WA in the previous section.

7.2.1 Cluster Size VS PPR Using Sequential LBA

Calculated PPR values for sequential LBA transfers are shown in Table 7.3. The PPR values for each cluster size (Figure 7.3) show a similar trend to WA (Figure 7.1), but the PPR valued are smaller. This suggests that a greater TBW for sequential LBA transfers would be calculated using PPR caused by not including wear from erasing unused pages. Cluster sizes smaller than the NAND page size appeared to increase PPR.

Table 7.3 Cluster Size VS PPR Using Sequential LBA

Cluster Size VS PPR Using Sequential LBA				
Cluster Size	Open Ended		Predetermine	
	Type 1	Type 2	Type 1	Type 2
512	8.56	8.56	8.56	8.56
2048	2.57	3.57	2.57	3.57
4096	1.07	2.57	1.07	2.57
8192	1.07	1.82	1.07	1.82
16384	1.07	1.45	1.07	1.45
32768	1.07	1.26	1.07	1.26
65536	1.07	1.17	1.07	1.17

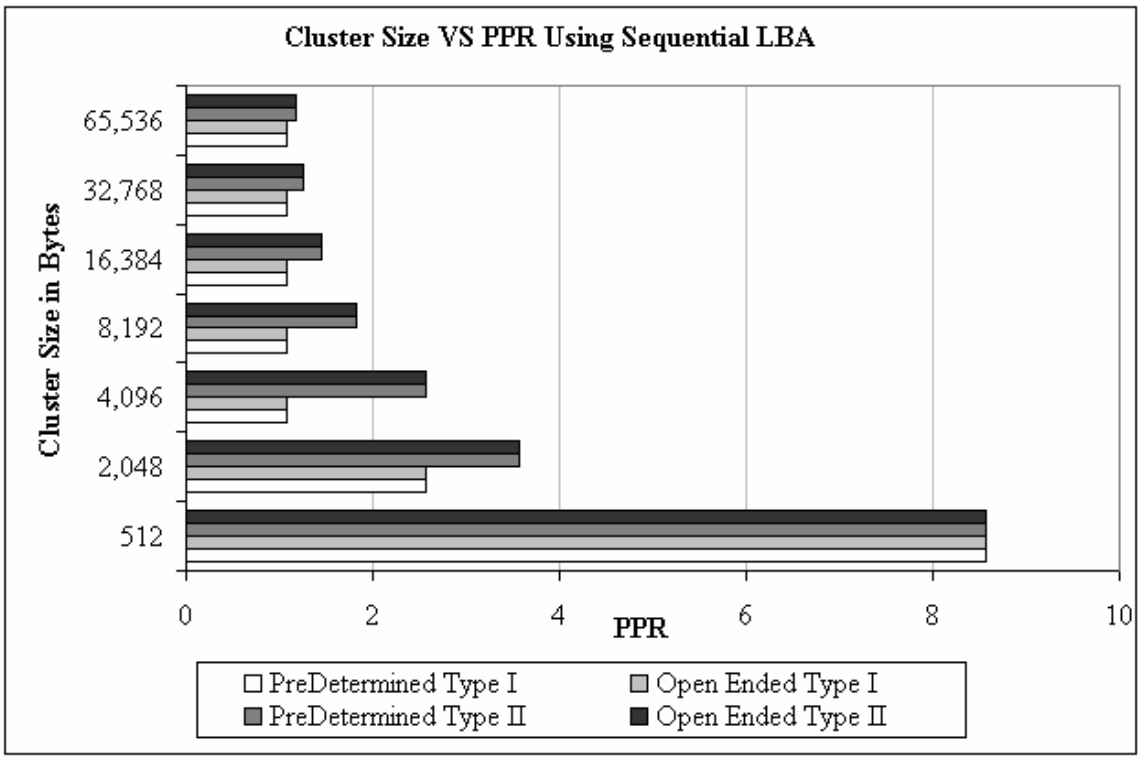

Figure 7.3 Cluster Size VS PPR Using Sequential LBA

7.2.2 Cluster Size VS PPR Using Random LBA

PPR calculated for random LBA transfers for each cluster size and for Type I and Type II cluster sizes are shown in Table 7.4. The PPR values for each cluster size (Figure 7.4) show a similar trend to the WA values (Table 7.2) for random LBA addressing. Smaller cluster sizes during each transfer type generated a greater PPR, but the value is smaller than the WA. This would suggest using PPR in calculating TBW for random LBA transfers may not be as accurate because of not including erased pages as part of wear. Cluster sizes smaller than the NAND page size appeared to increase PPR significantly.

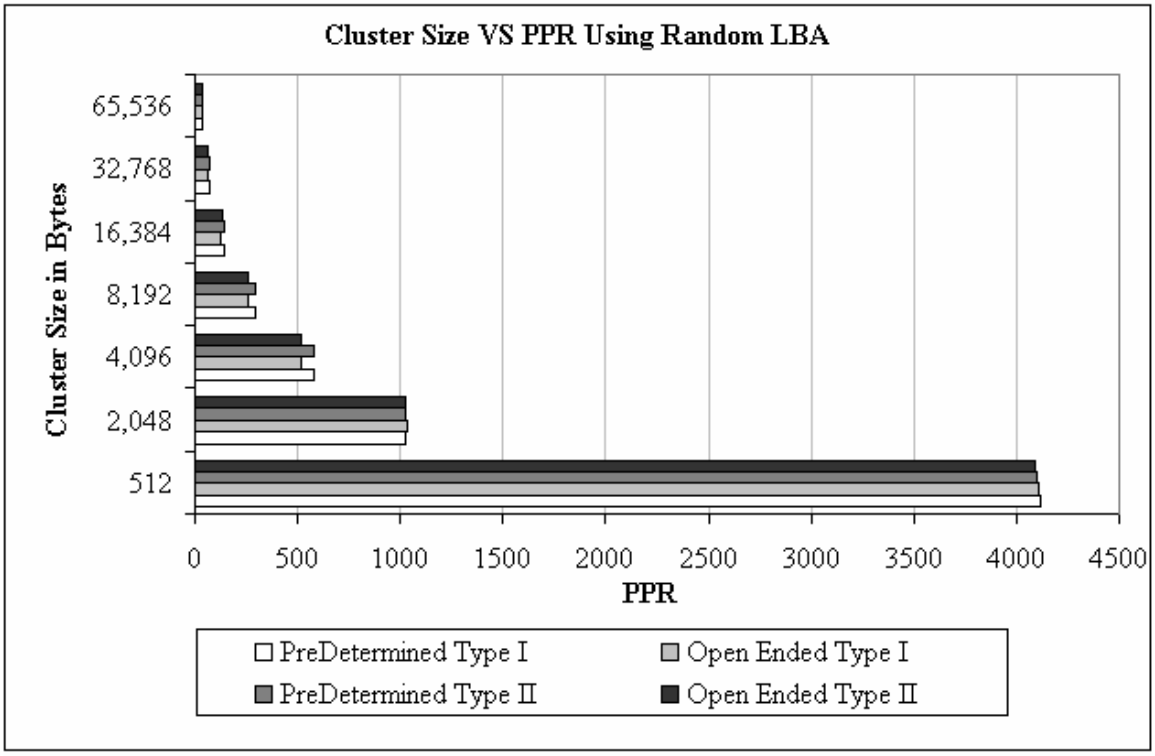

Figure 7.4 Cluster Size VS PPR Using Random LBA

Table 7.4 Cluster Size VS PPR Using Random LBA

Cluster Size VS PPR Using Random LBA				
Cluster Size	Open Ended		Predetermine	
	Type 1	Type 2	Type 1	Type 2
512	4112.30	4092.00	4115.99	4096.91
2048	1029.29	1027.91	1028.37	1027.81
4096	514.37	514.68	580.08	281.41
8192	257.18	259.04	289.94	291.66
16384	128.56	130.59	144.98	146.87
32768	64.79	66.45	72.99	74.54
65536	32.90	34.30	36.98	38.20

7.3 Measurement of Initial (new) WA, IOPS and PER Performance

The MMC device used for this thesis was brand new (unwritten). The full LBA range of the MMC device was first sequentially written to initialize all sectors and spare block area. Using the maximum sequential cluster size of 64 KBytes, the PER for the device was measured at 127/128 (Ideal 128/128). The IOPS performance was measured at 8.3 MBytes/sec. The calculated WA using equation (4.2) was 1.01. Once the MMC device was completely written, a second identical transfer was performed. A PER of 58 was observed (58/128). The IOPS performance decreased to 7.6 MBytes/sec, or 8% slower. The calculated WA increased to 2.34. The data is shown in

Table 7.5 and may suggest the device does not begin to wear level significantly until all NAND blocks including spare block area are used. The three measurements are compared in Figure 7.5. Using the initial (new) WA and PPR values may suggest inaccurate device longevity calculations.

Table 7.5 PER, IOPS and WA Measured Unwritten (new) VS Written (used)

PER		IOPS		WA (equation 4.2)	
PER New	PER Used	IOPS New	IOPS Used	WA (New)	WA (Used)
127	58	8.3 Byte/sec	7.6 Byte/sec	1.01	2.34

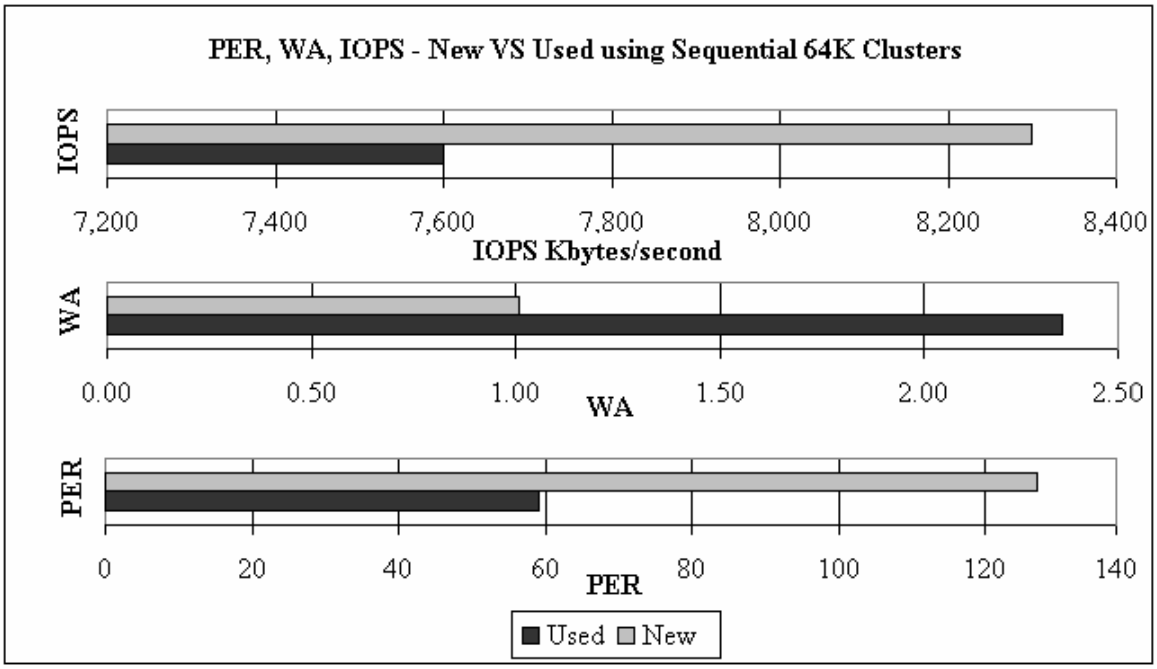

Figure 7.5 PER, IOPS and WA Measured Unwritten (new) VS Written (used)

7.4 Measurement of Input Output Per Second (IOPS)

Input Output Per Second (IOPS) was measured during each test flow. Data was collected to suggest a relationship between longevity (WA) and IOPS performance.

7.4.1 Cluster Size VS IOPS Using Sequential LBA

IOPS performance was observed to increase with cluster size on sequential transfers (see Table 7.6). Figure 7.6 compares the cluster size to IOPS for both Type I and Type II tests. The average IOPS write performance improves with full device transfers. The WA of Table 7.1 follows this trend. This may suggest that greater IOPS performance is an indicator of less WA overhead for sequential LBA transfers.

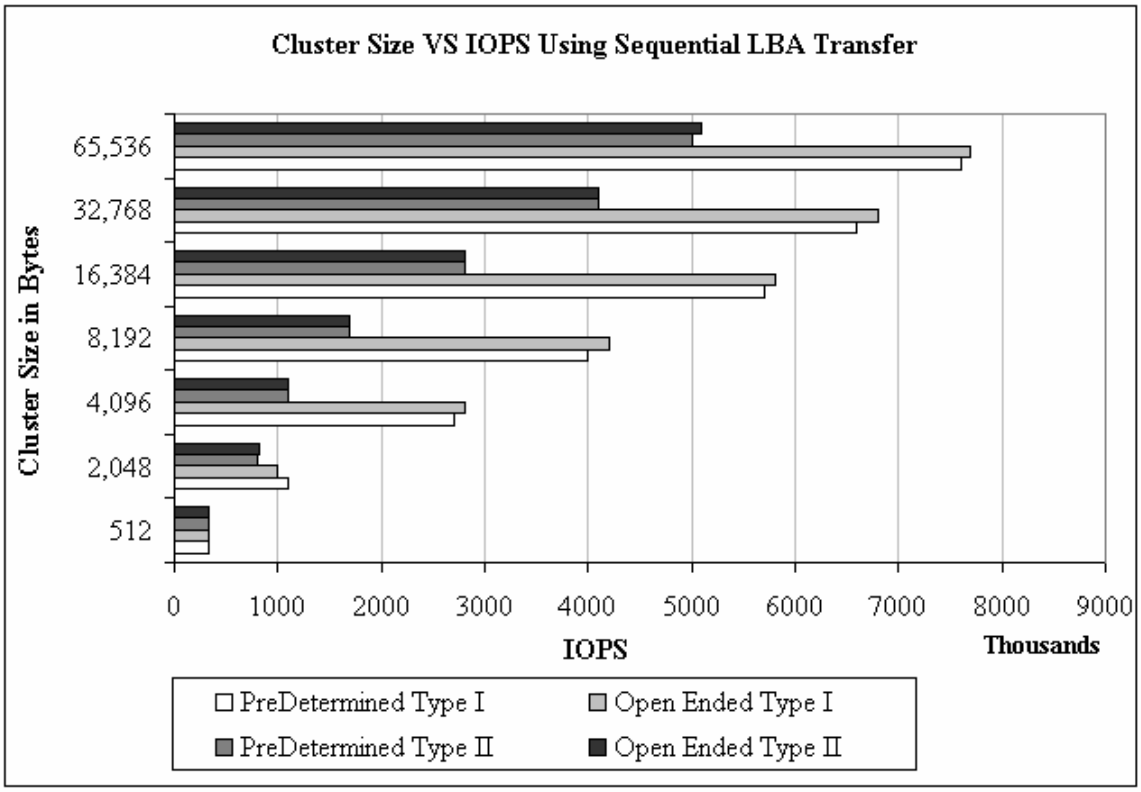

Figure 7.6 Cluster Size VS IOPS Using Sequential LBA

Table 7.6 Cluster Size VS IOPS Using Sequential LBA

Cluster Size VS IOPS (KBytes/sec) Using Sequential LBA				
Cluster Size	Open Ended		Predetermine	
	Type 1	Type 2	Type 1	Type 2
512	324	326.4	327.5	331.4
2048	1000	821.8	1100	804.8
4096	2800	1100	2700	1100
8192	4200	1700	4000	1700
16384	5800	2800	5700	2800
32768	6800	4100	6600	4100
65536	7700	5100	7600	5000

7.4.2 Cluster Size VS IOPS Using Random LBA

The IOPS performance increased with larger clusters on random LBA transfers (see Table 7.7). The performance was considerably slower than sequential LBA transfers.

Figure 7.7 shows the IOPS performance of random LBA addressing for each cluster size. The WA of Table 7.2 shows a similar trend. This may suggest a relationship between IOPS and WA overhead for random LBA transfers.

Table 7.7 Cluster Size VS IOPS Using Random LBA

Cluster Size VS IOPS (KBytes/sec) Using Random LBA				
Cluster Size	Open Ended		Predetermine	
	Type 1	Type 2	Type 1	Type 2
512	2.4	2.4	2.4	2.4
2048	9.5	9.6	9.5	9.7
4096	19.3	19.1	16.8	17.5
8192	38.5	38.6	33.9	34.1
16384	77.1	76.3	67.7	66.7
32768	152.1	149.5	134.2	133.5
65536	296.9	283.1	262.6	252.7

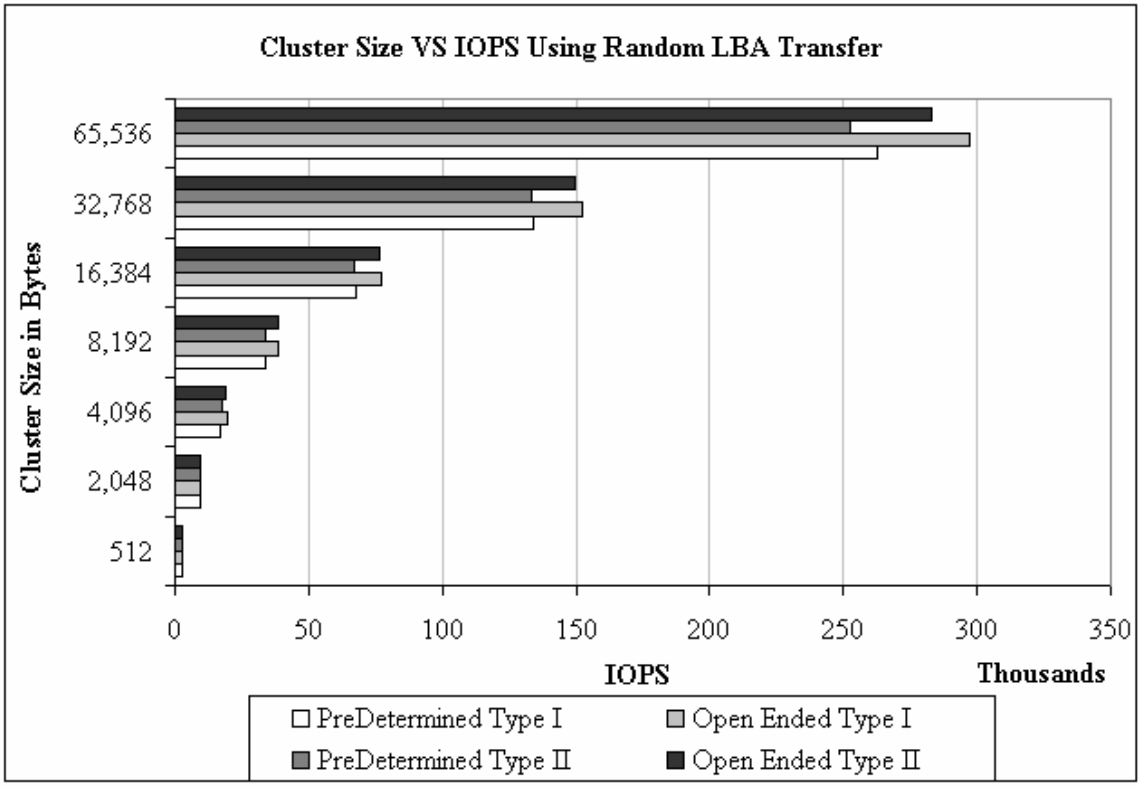

Figure 7.7 Cluster Size VS IOPS Using Random LBA

7.5 Measurement of Terabytes Written (TBW)

Using the measured values of WA and PPR, the calculated TBW capacity of the MMC device is compared in the following sections.

7.5.1 TBW Using WA

The Terabytes Written (TBW) capacity of the MMC device can be calculated using equation 5.3 and WA from Tables 7.1 and 7.2. Figure 7.8 shows the TBW using random and sequential LBA transfers for Type I and Type II tests and each cluster size. The data is scaled to Gigabytes Written (log scale has been used in x-axis). The data (Table 7.8) shows sequential LBA transfers of 4 KBytes or larger clusters produce similar TBW results. This may suggest the controller was optimized for sequential 4 KByte clusters. Random LBA transfers produced significantly smaller TBW capacity.

Table 7.8 TBW (Gigabytes) VS Cluster Size Using WA and Type I Test

TBW (Gigabytes) VS Cluster Size With WA and Type I Test				
Cluster Size	Open Ended		Predetermine	
	Random	Sequential	Random	Sequential
512	2.5	930.2	2.5	930.1
2048	10	4961.6	10	4965.0
4096	20.1	17662.4	17.8	17619.1
8192	40.1	17616.1	35.7	17619.9
16384	80.2	17654.6	71.3	17619.9
32768	160.1	17623.8	142.7	17631.5
65536	318.9	17612.2	285.5	17604.5

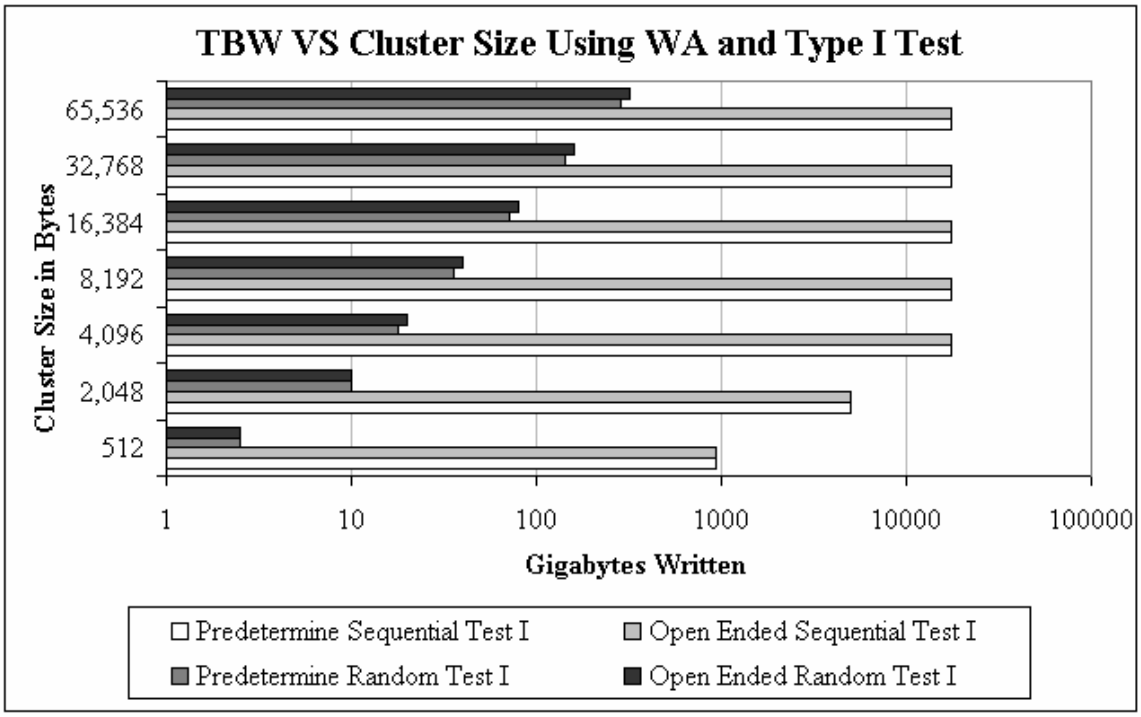

Figure 7.8 TBW (Gigabytes) VS Cluster Size Using WA and Type I Test

7.5.2 TBW Using PPR

The Terabytes Written (TBW) of the MMC device can also be calculated using equation 5.3 and measured PPR. Table 7.9 shows the TBW for random and sequential Type I transfers for each cluster size. The results are compared in Figure 7.9 and scaled to Gigabytes Written (log scale has been used in x-axis). A doubled TBW rating was observed using PPR compared to WA (Table 7.8). This may suggest an inaccurate TBW calculation (double) using PPR caused by including the wear of erasing unused pages.

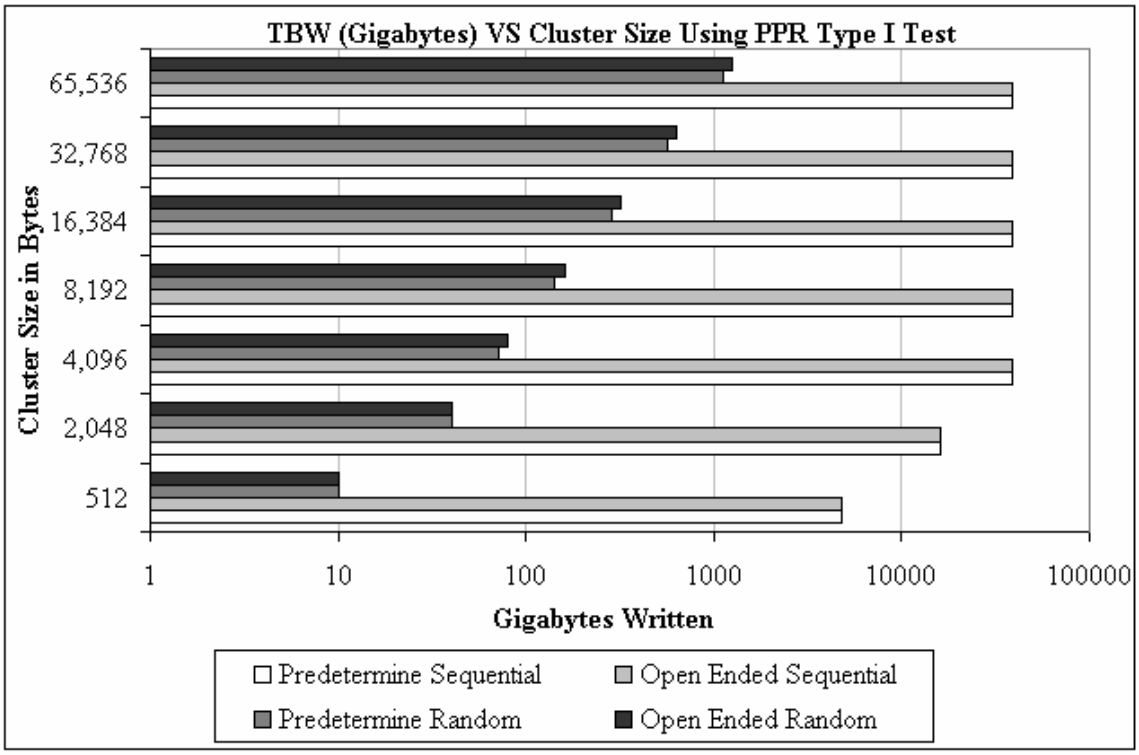

Figure 7.9 TBW (In Gigabytes) VS Cluster Size Using PPR and Type I Test

Table 7.9 TBW (In Gigabytes) VS Cluster Size Using PPR and Type I Test

TBW (Gigabytes) VS Cluster Size Using PPR and Test I Test				
Cluster Size	Open Ended		Predetermine	
	Random	Sequential	Random	Sequential
512	10.0	4804.3	10.0	4804.3
2048	40.0	16002.0	40.0	16002.0
4096	80.0	38434.7	70.9	38434.7
8192	159.9	38434.7	141.8	38434.7
16384	319.9	38434.7	283.7	38434.7
32768	634.7	38434.7	563.4	38434.7
65536	1250.0	38437.7	1112.1	38437.7

7.5.3 TBW and IOPS

Shown in Figure 7.10 is the relationship between TBW to IOPS performance for Type I test using random and sequential LBA transfers (log scale has been used in x-axis). The TBW data is scaled to Gigabytes (see Table 7.10). Cluster sizes greater than

4K during sequential transfers show small increases in IOPS performance. This may suggest limitations of the MMC interface clock frequency the tester used for the analysis. The graph suggests that slower IOPS performance represents greater WA. This may be used to generalize how longevity may be represented by IOPS without direct observation.

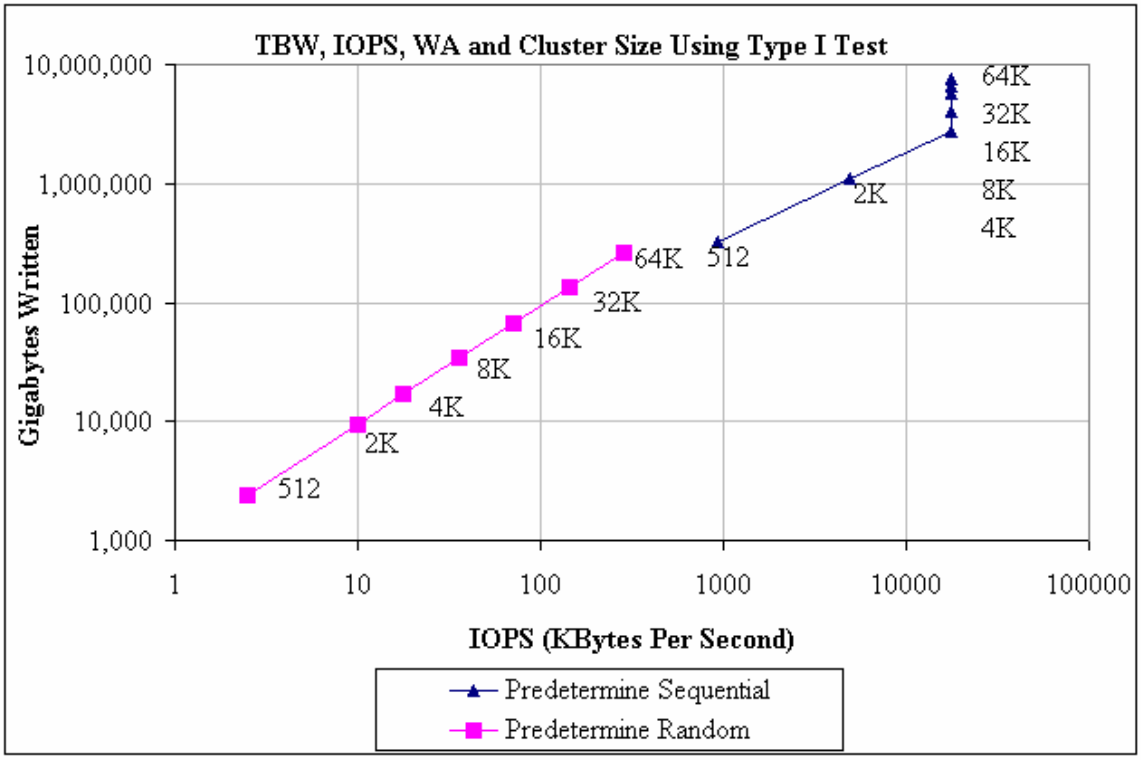

Figure 7.10 TBW VS IOPS VS Cluster Size Using WA and Type I Test

Table 7.10 TBW VS IOPS VS Cluster Size Using WA and Type I Test

TBW VS IOPS VS Cluster Size Using WA and Type I				
Cluster Size	IOPS (KBytes/sec)		TBW (Gigabytes)	
	Random	Sequential	Random	Sequential
512	2.4	328	2.5	930.1
2048	9.5	1100	10.0	4965.0
4096	16.8	2700	17.8	17619.9
8192	33.9	4000	35.7	17619.9
16384	67.7	5700	71.3	17619.9
32768	134.2	6600	142.7	17631.5
65536	262.6	7600	285.5	17604.5

7.5.4 TBW Comparison of WA and PPR

The TBW for both methods for each cluster size is shown in Table 7.11. They are compared in Figure 7.11 to the ideal TBW of equation 5.1 (log scale has been used in x-axis). The results suggest that by not including the wear caused by erasing unused pages, PPR results in a doubling of TBW. Sequential transfers of 4 KByte or larger clusters appear to show TBW (PPR) approaching the ideal value of equation 5.1.

Table 7.11 Comparing TBW Using WA and PPR VS Cluster Size For Type I Test

TBW (Gigabytes) from WA and PPR and Type I Test				
Cluster Size	TBW (WA)		TBW (PPR)	
	Random	Sequential	Random	Sequential
512	2.5	930.1	10.0	4804.3
2048	10.0	4965.0	40.0	16002.0
4096	17.8	17619.9	70.9	38434.7
8192	35.7	17619.9	141.8	38434.7
16384	71.3	17619.9	283.7	38434.7
32768	142.7	17631.5	563.4	38434.7
65536	285.5	17604.5	1112.1	38434.7

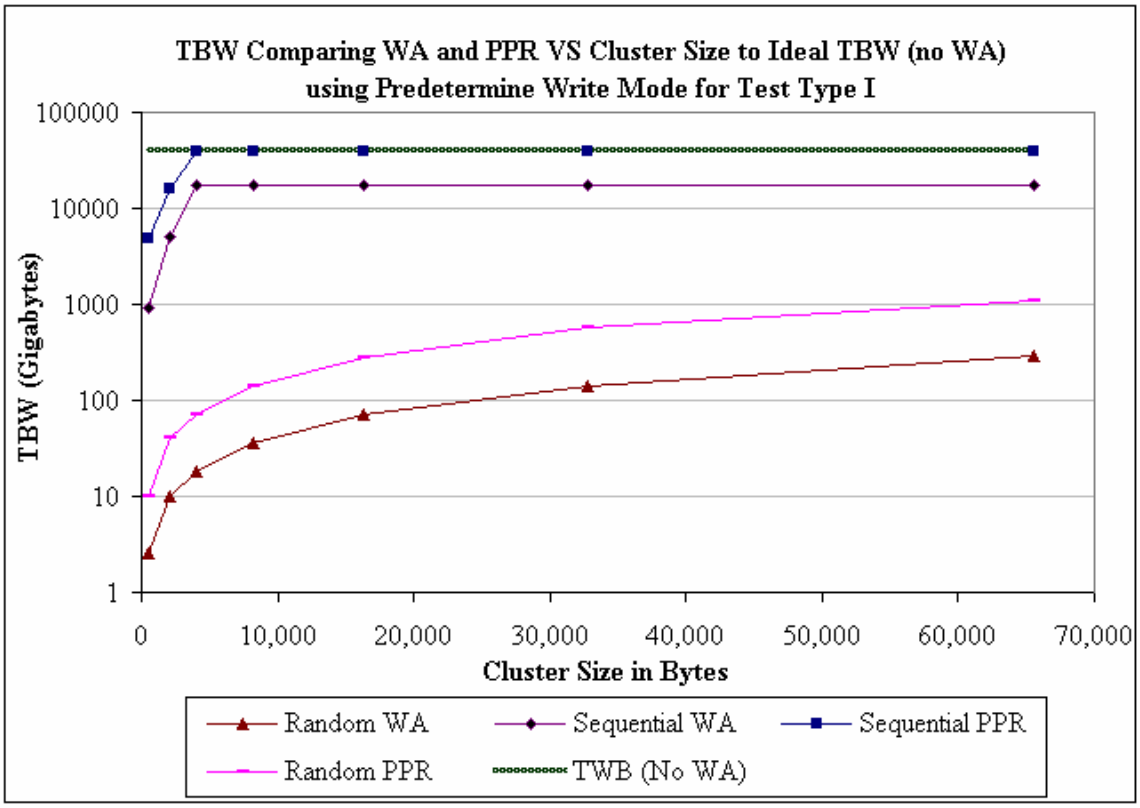

Figure 7.11 TBW Comparing WA VS Cluster Size Using Type I Test

CHAPTER 8 CALCULATING AND ESTIMATING END OF LIFE

The observed wear leveling data demonstrated how random LBA transfers and cluster sizes affect WA. Consider TBW at the 4 KByte cluster size (FAT-32). The calculation of TBW using WA (Table 7.11) appears to vary between 17 Gigabytes (random) and 17 Terabytes (sequential). This represents the extreme TBW values if using complete random or sequential LBA transfers over the life of the MMC device. However, the real life usage of flash device is somewhere in between. In this chapter, different combinations of random and sequential LBA transfers are used to determine endurance and end of life.

8.1 TBW Using a Ratio of Random and Sequential LBA Transfers

Depending upon the host application, the ratio of random and sequential LBA transfers over the life of the device may vary. Each type of LBA transfer is represented by a different WA constant. Consider calculating the TBW value as a summation of two parts. The first part is calculated using the number of random LBA transfers and WA constant. The second part is calculated using the number of the sequential LBA transfers and WA constant. Modified to reflect these changes, equation 5.3 yields equation 8.1 and shows the TBW for each transfer type and WA constant.

$$TBW = \frac{LDE(LBA)}{WA(random)} \times (\%Random) + \frac{LDE(LBA)}{WA(Sequential)} \times (\%Sequential) \qquad (8.1)$$

The TBW shown in Table 8.1 was calculated using equation 8.1 and the TBW from Table 7.11. The LDE value was calculated from equation 5.2. The data shows TBW for nine combinations of random to sequential LBA transfers. The table reflects an earlier range between 17 Gigabytes (random) and 17 Terabytes (sequential) using WA. Take the case when the ratio of random and sequential LBA transfers is equal (50% each), the TBW for WA is 8.8 Terabytes. This is nearly half of the capacity compared to purely sequential LBA transfers. When considering PPR and the same equal ratio of random and sequential LBA transfers, TBW is reduced by half from 38.4 Terabytes to 19.2 Terabytes.

Table 8.1 shows us that depending on how the flash device is used (ratio of random versus sequential transfers) it will influence the amount of data that can be transferred in and out of the flash memory (for the life of the device).

Table 8.1 TBW (Gigabytes) Comparing WA and PPR

TBW Considering The Percentage of Random and Sequential Transfers			
Total Transfers		TBW (Gigabytes)	
% Random	% Sequential	WA	PPR
0	100	17620	38435
5	95	16740	36517
10	90	15860	34598
25	75	13219	38844
50	50	8819	19253
75	25	4418	9662
90	10	1778	3907
95	5	898	1989
100	0	17	71

8.2 Example of Longevity Using A Hypothetical File Storage Application

Let's consider a 4 Gigabytes flash storage device for a digital camera. The transfer model consists of writing 4000 clusters of 4 KByte size (FAT-32), representing one

16MByte transfer (one picture). The user completely writes all LBA addresses daily (4 Gigabytes, equivalent to 256 pictures). It is assumed that each file transfer consists of a 50% ratio of sequential to random cluster transfers to represent FAT-32 file fragmentation and table updates. The TBW capacity was shown as 8.819 Terabytes (Table 8.1). Using equation 8.2, the life capacity of the device is calculated at 5.6 years. Life may be extended by changing the frequency and duty cycle of the file transfers however.

$$\frac{8{,}819{,}000{,}000{,}000TBW(bytes)}{16777216(filesizeInBytes)\times 256(files/day)\times 365(days)} = 5.6Years \qquad (8.2)$$

For comparison, the life calculation was performed using PPR. The TBW data from Table 7.9 was calculated life expectancy at 12.2 years. (see equation 8.3). This was twice the life time compared to using the WA value.

$$\frac{19{,}253{,}000{,}000{,}000TBW(bytes)}{16777216(filesizeInBytes)\times 256(files/day)\times 365(days)} = 12.2Years \qquad (8.3)$$

The data indicates that the life rating of an MMC device is affected by the size and ratio of random to sequential write transfers and the method used to determine write amplification. The life calculations suggest that using WA (total erased bytes to written bytes) is more accurate compared to PPR (NAND pages programmed to data written).

CHAPTER 9 CONCLUSION AND FUTURE RESEARCH

9.1 Conclusions

The thesis attempts to answer the question, “How long will my flash storage devices last?” At the writing of this thesis, the MultiMediaCard interface specification appears to have no method of retrieving wear leveling information. This may suggest that it is impossible to query the MMC device to determine remaining longevity. The answer to the original question may remain unresolved. Consider what is more important, end of life estimations or data retention capability? More than simply a catastrophic or cascading failure point, the question becomes “how reliable is long term storage backup on managed NAND devices?”

Without direct observation or methods to retrieve wear information, at least two approaches exist on how to determine device longevity. The first approach uses supplied performance data, such as Write Amplification (WA) and Terabytes Written (TBW), from the manufacturer. However, the customer may either be unable to obtain this information or reproduce and verify accuracy when applied to their usage model. The second approach is to verify the device using a black box approach. By stressing the device writing their usage model, the customer performs distributed testing on the device and verifies data retention.

The thesis measures and predicts the longevity of a MMC device by direct observation of wear directly on the NAND die component. Two methods to represent Write Amplification have been presented. The first method considered the total bytes erased on the NAND flash to the total bytes written by the host. The observed wear leveling data suggested this to be the most accurate representation of WA. The second method considers the ratio of the total NAND pages programmed to the pages written by the host (PPR). This method does not consider the wear overhead of erasing unused pages during wear leveling. The results suggests that using PPR to calculate longevity may not be representative of all flash cell wear.

Device longevity is represented by TBW (Terabytes Written) capacity and considers the effects of Write Amplification. This measures the total number of bytes the device may accept before reaching end of life. Using TBW, equations to calculate device life (in years) were presented. The equations consider a usage model determined by a relationship between transfer frequency, percentage of write to read transfers and write duty cycle. The result of the life equations is to represent the longevity of the device, in years of use, until a one year data retention capability is reached.

In this thesis, to obtain WA information on a MMC device, direct observation of erase and program cycles (E/P) and pages programmed on the NAND die was performed. Two MMC write modes were used; open ended and predetermine. The measured data suggests that the MMC predetermine write mode has a minor WA advantage. A relationship was observed between the test platform MMC driver interface IOPS performance and WA. An increase in IOPS performance generally indicated a decrease in

WA. The results suggests that write data transfers less than the NAND page size, especially random LBA addressing, significantly reduced IOPS and increased WA.

The collected data was applied to a hypothetical file storage application usage model. The model consisted of 256 file transfers to completely write a 4 Gigabyte MMC device. Each file transfer consisted of writing 4000 4 KByte clusters. Each transfer has a 50 percentage ratio of random to sequential LBA addresses. Using the measured WA, life was calculated to be 5.6 years before end of life was reached. By measuring the ratio of random and sequential LBA addressing on the MMC interface for other usage models, the calculated WA values may be used to calculate the life of the device. This may be more accurate than using a single WA constant to represent (average) all combinations of LBA addressing ratios over the life of the device.

The results suggest that applications stressing the device, such as a daily file backup, may diminish longevity more quickly. However, as determined by this thesis, infrequent file storage use may approach the 10 year advertised rating. The thesis concludes that when used for file storage, a managed NAND device can be reliable, but perhaps only when long term file storage is not required. A larger storage device will have an increase in TBW, applying the same usage model may increase life. However, the results suggests that using the MMC device for non file storage application, such as embedded systems writing frequent small randomly addressed transfers, may wear the device quickly.

9.2 Future Research

At the writing of this thesis, new technologies in non volatile memory storage are in consideration. Floating gate technology may quickly be replaced by more efficient

methods. These new technologies may redefine methods to determine longevity and be more difficult to standardize.

To accurately determine longevity, a method to directly measure device wear on portable Managed NAND devices may be necessary. An alternative to direct measurement may be a software application monitoring transfer size and LBA addressing randomness on portable storage devices. The application may calculate life using WA values based upon standardized usage models from industry. The models may include file storage applications such as Camera, GPS, Cell phone, File backup, etc.

New protocol commands, called TRIM, are being supported on Intel© SSD products in conjunction with Microsoft Windows 7 operating system [21]. These commands address, and mostly eliminate, the performance penalty of wear leveling unwanted sectors of data. A method may be explored to extend the MMC interface specification to include TRIM commands on portable managed NAND devices.

The emerging MMC V4.4 specification allows dynamic allocation of storage, similar to partitioning. These new MMC commands may also introduce high reliability ‘boot’ sections using a combination of SLC and MLC modes. The definition how to measure WA and determine longevity may be investigated.

REFERENCES

[1] "NAND Flash Design and Use Considerations.", Micron Technology Technical Note, TN-29-17.fm, REV.A, August 2006, www.micron.com/products/nand/.

[2] Doug Kearns, "Practical Guide to Endurance and Data Retention", Application Note, Revision A, Amendment 0, September 16, 2005. http://www.spansion.com

[3] "Online course on Embedded Systems.", Module 14, EE Herald Magazine, http://www.eeherald.com/section/design-guide/esmod16.html.

[4] Roberto Bez, Emilio Camerlenghi, Alberto Modelli, and Angelo Visconti. "Introduction to Flash Memory", Invited Paper, Proceedings of The IEEE, Vol. 91, No. 4, April 2003

[5] Krieger, G. Swanson, R. M.,"Fowler-Nordheim electron tunneling in thin Si-SiO2-Al structures." Journal of Applied Physics, Volume 52 Issue 9, July 6, 2009. Standford Electronics Laboratories, Stanford, CA 94305. www.ieeexplore.ieee.org

[6] "The Evolving NANDscape", MLC vs. SLC Flash. www.micron.com/nandcom

[7] "SLC vs. MLC: An Analysis of Flash Memory", Whitepaper, Super Talent Technology, Inc. http://www.supertalent.com

[8] "NAND Evolution and its Effects on Solid State Drive (SSD) Useable Life", White Paper, Western Digital Technologies, Inc. 2009. http://www.wdc.com

[9] Arie Tal., "Two Technologies Compared: NOR vs. NAND", M-Systems Flash Disk Pioneers, Ltd., 2003, http://www.m-sys.com

[10] "VLSA and Embedded System Technical Library", Flash Memory, NOR and NAND Flash, http://vetechlib.blogspot.com/2008/09/flash-memory_16.html

[11] Kelly Hirsch, "Programming NAND devices", Technical Guide, Data I/O Corporation. http://www.datio.com/nand/nandflash.asp

[12] "Open NAND Flash Interface Specification" Revision 2.2, October 7 2009, ONFI Workgroup. www.onfi.org

[13] “Electrically Erasable Programmable Rom (Eeprom) Program/Erase Endurance and Data Retention Test:” JESD22-A117B, March 1 2009, JEDEC Solid State Technology Association. www.jedec.org

[14] JEDEC Standard., “Embedded MultiMediaCard(eMMC) eMMC/Card Product Standard, High Capacity, including Reliable Write, Boot, Sleep Modes, Dual Data Rate, Multiple Partitions Supports and Security Enhancement” JEDEC Standard No. 84-A44. JEDEC Publications Department, March 2009. JEDEC Solid State Technology Association.

[15] Peter Bright, “Why new hard disks might not be much fun for XP users”, The Microsoft ecosystem, March 10, 2010, Microsoft Inc, www.microsoft.com

[16] Scott Chen, “What Types of ECC Should Be Used on Flash Memory?”, Application Note, November 27, 2007. Spansion Inc., http://www.spansion.com

[17] Tatsuya Kawamatsu, “technology for managing NAND flash”, Hagiwara sys-com co., Ltd., http://techon.nikkeibp.co.jp

[18] “Longterm Data Endurance (LDE) for Client SSD”, White Paper For JEDEC 64.8 workgroup. October 31, 2008, www.SanDisk.com

[19] Jamon Bowen, “Understanding IOPS”, White Paper, Texas Memory Systems, July 2007, http://www.ramsan.com/whitepapers.htm

[20] IOMETER, IOmeter project homepage, http://www.iometer.org

[21] Frank Shu , “Windows 7 Enhancements for Solid-State Drivers”, WinHEX 2008, December-2008. COR-T558_Shu_Taiwan.pdf. Microsoft Corporation Inc.

APPENDIX

Test I and Test II Raw Data

TEST I	PreDetermined	PreDetermined					
TRAN SIZE	Num Transfers	LBA MODE	Tran Rate	Total Bytes Written	Time	ERASES	Page Program Counts
512	8,032,256	SEQUENTIAL	327.5 KBs	4,112,515,072	3 hours 24 min 40 secs 157 msec	693,624	17,185,040
2,048	2,008,064	SEQUENTIAL	1.1 MBs	4,112,515,072	1 hours 1 min 11 secs 391 msec	129,944	5,161,154
4,096	1,004,032	SEQUENTIAL	2.7 MBs	4,112,515,072	0 hours 24 min 33 secs 316 msec	36,616	2,156,319
8,192	502,016	SEQUENTIAL	4.0 MBs	4,112,515,072	0 hours 16 min 5 secs 403 msec	36,616	2,157,343
16,384	251,008	SEQUENTIAL	5.7 MBs	4,112,515,072	0 hours 11 min 33 secs 743 msec	36,616	2,157,343
32,768	125,504	SEQUENTIAL	6.6 MBs	4,112,515,072	0 hours 9 min 48 secs 226 msec	36,592	2,156,291
65,536	62,752	SEQUENTIAL	7.6 MBs	4,112,515,072	0 hours 8 min 38 secs 288 msec	36,648	2,158,343
512	8,032,256	RANDOM	2.4 KBs	5,120,000	EST 19 days 20 hours	320,086	10,289,968
2,048	2,008,064	RANDOM	9.5 KBs	20,480,000	EST 5 days 15 min	319,868	10,283,680
4,096	1,004,031	RANDOM	16.8 KBs	4,112,510,976	66 hours 17 min 47 secs 232 msec	36,193,060	1,164,836,108
8,192	502,015	RANDOM	33.9 KBs	4,112,506,880	32 hours 53 min 54 secs 885 msec	18,091,974	582,210,678
16,384	251,007	RANDOM	67.7 KBs	4,112,498,688	16 hours 28 min 10 secs 557 msec	9,044,922	291,126,085
32,768	125,504	RANDOM	134.2 KBs	4,112,515,072	8 hours 18 min 39 secs 960 msec	4,521,212	146,566,711
65,536	62,751	RANDOM	262.6 KBs	4,112,449,536	4 hours 14 min 51 secs 633 msec	2,259,984	74,258,191
TEST I	Open Ended						
512	8,032,256	SEQUENTIAL	0.324 MBs	4,112,515,072	3 hours 26 min 34 secs 471 msec	693,600	17,182,992
2,048	2,008,064	SEQUENTIAL	1.0 MBs	4,112,515,072	1 hours 2 min 19 secs 966 msec	130,032	5,166,274
4,096	1,004,032	SEQUENTIAL	2.8 MBs	4,112,515,072	0 hours 23 min 37 secs 659 msec	36,528	2,151,199
8,192	502,016	SEQUENTIAL	4.2 MBs	4,112,515,072	0 hours 15 min 34 secs 918 msec	36,624	2,157,343
16,384	251,008	SEQUENTIAL	5.8 MBs	4,112,515,072	0 hours 11 min 18 secs 9 msec	36,544	2,152,223
32,768	125,504	SEQUENTIAL	6.8 MBs	4,112,515,072	0 hours 9 min 39 secs 523 msec	36,608	2,156,319
65,536	62,752	SEQUENTIAL	7.7 MBs	4,112,515,072	0 hours 8 min 32 secs 178 msec	36,632	2,157,925
512	8,032,256	RANDOM	2.4 KBs	5,120,000	EST 19 days 20 hours	319,824	10,280,762
2,048	2,008,064	RANDOM	9.5 KBs	20,480,000	EST 5 days 15 min	320,212	10,292,889
4,096	1,004,032	RANDOM	19.3 KBs	4,112,510,976	57 hours 50 min 13 secs 552 msec	32,144,954	1,032,894,682
8,192	502,015	RANDOM	38.5 KBs	4,112,506,880	28 hours 57 min 23 secs 975 msec	16,080,018	516,426,829
16,384	251,007	RANDOM	77.1 KBs	4,112,515,072	14 hours 28 min 38 secs 582 msec	8,044,574	258,148,343
32,768	125,503	RANDOM	152.1 KBs	4,112,482,304	7 hours 20 min 5 secs 693 msec	4,030,868	130,108,569
65,536	62,751	RANDOM	296.9 KBs	4,112,449,536	3 hours 45 min 25 secs 489 msec	2,022,796	66,054,592
INITIALIZE							
65,536	62,752	FORWARD	8.3 MBs	4,112,515,072	0 hours 7 min 53 secs 896 msec	15,784	2,010,133

TEST II	PreDetermined						
TRAN SIZE	Num Transfers	LBA MODE	ansfer Rate	Total Bytes Written	Time	ERASES	Page Program Counts
512	2,409,677	SEQUENTIAL	331.4 KBs	1,233,754,624	1 hours 0 min 35 secs 796 msec	208,140	5,159,194
2,048	602,419	SEQUENTIAL	804.8 KBs	1,233,754,112	0 hours 24 min 56 secs 988 msec	67,160	2,150,304
4,096	301,209	SEQUENTIAL	1.1 MBs	1,233,747,968	0 hours 17 min 44 secs 91 msec	38,980	1,549,056
8,192	150,604	SEQUENTIAL	1.7 MBs	1,233,747,968	0 hours 11 min 45 secs 993 msec	24,852	1,096,873
16,384	75,302	SEQUENTIAL	2.8 MBs	1,233,747,968	0 hours 7 min 3 secs 458 msec	17,888	874,809
32,768	37,651	SEQUENTIAL	4.1 MBs,	1,233,747,968	0 hours 4 min 46 secs 457 msec	14,424	760,847
65,536	18,825	SEQUENTIAL	5.0 MBs	1,233,715,200	0 hours 3 min 51 secs 784 msec	12,600	705,724
512	2,409,677	RANDOM	2.4 KBs	5,120,000	EST 5 Day 22 hours 47 min	317,568	10,242,274
2,048	602,419	RANDOM	9.7 KBs	20,480,000	EST 1 Day 11 hours 19 Min	318,368	10,278,076
4,096	301,209	RANDOM	17.5 KBs	40,960,000	EST 19 hours 35min	360,270	11,628,187
8,192	150,604	RANDOM	34.1 KBs	1,233,747,968	9 hours 48 min 4 secs 400 msec	5,433,856	175,702,913
16,384	75,302	RANDOM	66.7 KBs	1,233,747,968	5 hours 0 min 51 secs 581 msec	2,726,928	88,476,814
32,768	37,651	RANDOM	133.5 KBs	1,233,747,968	2 hours 30 min 23 secs 179 msec	1,374,352	44,905,459
65,536	18,825	RANDOM	252.7 KBs	1,233,715,200	1 hours 19 min 27 secs 944 msec	695,272	23,014,302
TEST II	Open Ended						
512	2,409,677	SEQUENTIAL	326.4 KBs	1,233,754,624	1 hours 1 min 30 secs 890 msec	208,132	5,158,172
2,048	602,419	SEQUENTIAL	821.8 KBs	1,233,754,112	0 hours 24 min 26 secs 58 msec	67,144	2,149,280
4,096	301,209	SEQUENTIAL	1.1 MBs	1,233,752,064	0 hours 17 min 43 secs 353 msec	38,988	1,550,078
8,192	150,604	SEQUENTIAL	1.7 MBs	1,233,747,968	0 hours 11 min 51 secs 344 msec	24,852	1,096,874
16,384	75,302	SEQUENTIAL	2.8 MBs	1,233,747,968	0 hours 7 min 0 secs 667 msec	17,840	871,738
32,768	37,651	SEQUENTIAL	4.1 MBs	1,233,747,968	0 hours 4 min 45 secs 344 msec	14,316	759,241
65,536	18,825	SEQUENTIAL	5.1 MBs	1,233,715,200	0 hours 3 min 51 secs 420 msec	12,582	703,254
512	2,409,677	RANDOM	2.4 KBs	5,120,000	EST 5 Day 22 hours 47 min	317,192	10,230,006
2,048	602,419	RANDOM	9.6 KBs	20,480,000	EST 1 Day 11 hours 42 min	318,400	10,279,092
4,096	301,209	RANDOM	19.1 KBs	40,960,000	EST 17 hours 57 min	318,528	10,293,674
8,192	150,604	RANDOM	38.6 KBs	1,233,747,968	11 hours 55 min 11 secs 656 msec	4,820,228	156,051,456
16,384	75,302	RANDOM	76.3 KBs	1,233,747,968	4 hours 23 min 19 secs 59 msec	2,420,604	78,668,400
32,768	37,651	RANDOM	149.5 KBs	1,233,747,968	2 hours 14 min 18 secs 69 msec	1,222,428	40,031,549
65,536	18,825	RANDOM	283.1 KBs	1,233,715,200	1 hours 10 min 55 secs 195 msec	621,982	20,663,023