

# USING FLASH MEMORIES AS SIMO CHANNELS FOR EXTENDING THE LIFETIME OF SOLID-STATE DRIVES

*Maria Varsamou, and Theodore Antonakopoulos*

Department of Electrical and Computers Engineering,  
University of Patras, 26500, Rio-Patras, Greece  
E-mail: mtvars, antonako@upatras.gr

## ABSTRACT

Reliability and I/O performance are the two basic metrics that determine the quality of solid-state drives (SSDs), especially in enterprise storage systems. Flash memories, the most popular non-volatile memory used in today's solid-state drives, demonstrate a time-varying behavior in terms of raw bit errors per program/erase cycle. This paper presents experimental results regarding the time-varying behavior as well as the statistical characteristics of single and multiple level cell flash memories. A new method that exploits these characteristics and uses the flash memories as Single Input Multiple Output channels for extending the lifetime of storage devices based on single level cell technology is presented. The method's efficiency is highlighted and its effect on the system's I/O performance is discussed.

*Index Terms*— Flash memory, solid-state drives, BCH.

## 1. INTRODUCTION

Solid-state drives (SSDs) have become a mature solution for consumer and enterprise storage systems. As storage medium, they use flash memories and they have to demonstrate similar or even better performance compared to the widely-used magnetic disks. The main constraints to the attributes of a SSD are imposed by the physical characteristics and the sizing limitations of the flash technology. Reliability and I/O performance are the two basic metrics evaluated for determining the quality of SSDs, especially in enterprise storage systems. These two parameters are mainly affected by the non-volatile technology used, the supported workload, the architecture and the functionality of the SSD's controller.

Regarding reliability, the highest performance is shown at the beginning of the lifetime of the device, due to the fact that flash memories demonstrate a time-varying behavior in terms of raw bit errors per program/erase cycle. The performance also drops when a capacity related limit for the number of stored pages is exceeded and is further reduced when the endurance cycles of the used flash chips are consumed. Endurance is defined as the number of program/erase cycles that can be performed before the SSD wears out. In order to

extend the endurance of the flash chips and to improve the reliability of the SSDs, error correction codes (ECC) are used [1]. On the other hand, the decoding procedures activated in cases of errors have an impact on the I/O performance of the SSDs. In this work we investigate the effect of the used flash technology on the endurance of a SSD and we propose a method that improves it by exploiting the statistical behavior of the flash channel.

Since flash-based storage devices are prone to random errors, binary BCH codes are usually used in SSDs. It is well known that the performance of these codes can be improved significantly when errors-and-erasures decoding is used, where an erasure indicates that this symbol has been associated with a high probability of being in error. Since in BCH codes a symbol corresponds to a single bit, a correct erasure is equivalent to recovering the corrupted bit. However, the performance of the decoder deteriorates seriously when the erasures are not set correctly. Therefore, a mechanism for producing highly reliable erasures is needed. Such a mechanism is presented in this work.

Specifically, in Section 2 flash technology is introduced and its reliability is discussed. Section 3 highlights the statistical characteristics of SLC and MLC NAND flash technologies using experimental results, while Section 4 presents the proposed method and demonstrates its advantage for extending the endurance of flash memories. Finally Section 5 discusses how the proposed method is implemented in a SSD and how it affects its I/O performance.

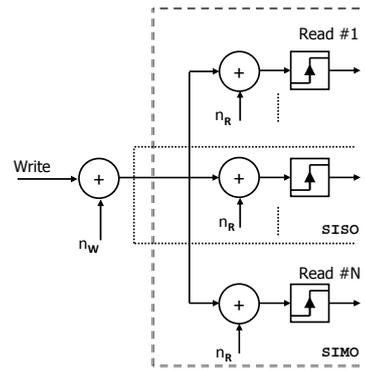
## 2. FLASH MEMORY TECHNOLOGY

NAND flash dimensions (nowadays below 40 nm) result to high storage density, making the design of high-capacity SSDs a feasible solution for consumer and enterprise storage systems. The NAND flash cell is a floating-gate transistor with two overlapping gates, one of them isolated in oxide. The IV characteristics of a flash cell are altered by controlling the voltages applied to the transistor gate, drain, source and bulk terminals. NAND cells are connected in series, forming blocks of pages. The operations related to a set of flash cells

is write (programming), read and erase. Fowler-Nordheim (FN) tunneling is used during NAND flash programming. A high voltage is applied to the cell gate, while drain, source and bulk are grounded. This process requires a very small current per cell, allowing many cells to be programmed at a time. Erase is performed at a block level, affects multiple pages and is also based on FN tunneling [2]. Reading is based on a charge integration mode and exploits the bitline parasitic capacitances. The charge stored in the floating gate determines the discharging process. The high electric field applied to the tunnel oxide during programming and erase results to the loss of its insulating properties. Variations on the stored charge are related with oxide aging, due to charge trapping within the oxide. Variations either on the stored charge or on the detected voltage during reading result to corrupted bits.

There are two types of NAND flash memories: SLC (single-level cell) that store a single bit per cell and MLC (multiple-level cell) that store two or more bits per cell. The typical endurance of a SLC cell is 100K ‘program and erase cycles’ (P/E), while for MLC endurance decreases to 10K P/E cycles. For enterprise storage systems, there are MLC memories with 30K erase/write cycles, but with longer erase/read/write times. MLC memory cells wear out faster due to physical changes in the dielectric (tunnel oxide) of the floating gate and experience read errors with higher probability due to variations in the threshold voltage at the control gate and the shorter voltage difference between adjacent levels. The lifetime of a SSD is determined by the P/E cycles of the complete flash memory (also called write endurance), the capacity of the SSD, the usage profile and the type of data, hot or cold. Write requests are classified into either hot or cold based on update frequency. As usage profile, we define the sequential/random characteristics of the write commands, the applied workloads, the I/O rate etc. Several techniques are applied for extending the SSDs’ lifetime, such as wear leveling, garbage collection, write amplification and over-provisioning, but their effect is beyond the scope of this work.

The NAND ICs used in SSDs contain a huge number of flash memory cells organized in arrays, analog circuits to erase/read/program the cells and hard detection circuits to recover the originally stored digital information. Therefore, the SSD controller that applies the error correction mechanisms, has to be based only on hard-decoded information. Up to now, a user page was encoded prior to being stored in the flash IC, and in case of corrupted bits, error correction was performed on the retrieved data, or higher level data recovery methods (like RAID) were used for regenerating the initially stored data. Using experimental results, it can be shown that the errors detected when a page is retrieved, can be categorized as permanent and as temporary. Some bit errors are related with the writing process and the aging of the flash cell, while others are also related with the variations of the



**Fig. 1.** The Flash memory as a SIMO channel.

reading process. The flash channel is usually modeled as a Single Input Single Output (SISO) channel and the noise is introduced by two independent sources, as it is shown in Figure 1. The second noise source is related only with the reading process and is also time-varying.

This work deals with the problem of enhancing the reliability of storage devices by improving the performance of BCH codes used in flash memories, and thus extending the maximum number of P/E cycles, for a given *user BER* and coding efficiency. The basic idea is that, since some error bits are related with the reading process, if we read the same page multiple times and associate the locations of all symbols with error probabilities, then we can detect some of the reading-related error bits, mark them as erasures and improve the error correction capability of the used ECC code without using any additional parity symbols. In this case, the flash channel is equivalent to a Single Input Multiple Output (SIMO) channel and results to better error conditions. An additional advantage is that due to the erasing mechanism used in flash memories and to the unequal binary error probabilities of the flash channel (ones may change to zeros and not vice versa in SLC valid blocks), the above mentioned process has small complexity, it can be implemented in hardware and has small or negligible effect on the SSD’s I/O performance.

### 3. FLASH CHARACTERIZATION

Although there is a large number of publications concerning the physics and the performance of flash memories, there is a lack of experimental data concerning the performance of flash ICs, probably due to the continuous progress of the flash technology. Therefore, we developed an experimental system for loading flash memories with different workloads and data patterns, in order to study their time-varying behavior and to collect information concerning the error conditions. In each experiment a block of pages is erased, then random data patterns are stored in all pages and finally all pages are read back and compared with the originally stored data. Each

page is read multiple times (up to five for SLC and up to fifteen for MLC). The complete experiment was executed on an embedded platform and most of the processing was performed at a local PowerPC along with dedicated hardware accelerators. Due to the large number of P/E cycles supported by SLC, the duration of each experiment per block takes almost a week, while for MLC memories just a couple of days.

In a system where error correction codes are used, two types of bit error ratios (BER) are defined. *Raw BER* is the BER experienced when the data are retrieved from the memory cells, while *user BER* is the BER experienced after applying error correction. In a typical hard disk drive, the *user BER* is specified as  $10^{-15}$ . Furthermore, the same *user BER* has to be guaranteed by a SSD for its total lifetime. For SLC memories, *raw BER* in the range of  $10^{-9}$  to  $10^{-11}$  has been reported for the first 100K P/E cycles (this is the endurance specified by the manufacturer), while for MLC memories the *raw BER* is in the range of  $10^{-5}$  to  $10^{-7}$  for the first 10K P/E cycles. For satisfying the *user BER* requirements, binary BCH codes are usually used in NAND flash memories due to their advanced coding efficiency and the existence of flexible and effective hardware circuits of acceptable complexity for their implementation [3]. In SSDs that use SLC memories, the minimum required ECC is 2-bits error correction per 512 bytes of user data, while for MLC a minimum of 6-bits error correction code is needed per 512 bytes of user data.

When the manufacturer-specified endurance cycles are consumed, the raw BER increases exponentially, as it is shown in Figs. 2 and 3. For extending the endurance period of a flash memory, a stronger ECC has to be used, but that requires more parity symbols for the same amount of user data. Flash memories use a fixed page size, and in each page user data (sectors), metadata and parity symbols are stored. Currently, the typical page size for both types of memory is 4320 bytes. That imposes an upper limit to the maximum number of parity symbols that can be used per sector. The use of a stronger code means that a smaller number of user sectors can be stored per page. As a consequence, the SSD's storage efficiency decreases.

The red curve in Fig. 2 and the curve in Fig. 3 show the experimental raw BER performance of SLC and MLC memories respectively. In both cases, experimental results have been collected for ten times the P/E cycles of the respective flash technology. Concerning the SLC memory, in order to extend the endurance to 300K cycles, a stronger BCH code has to be used to confront the  $4 \times 10^{-5}$  *raw BER*. In this case a BCH code able to correct 8-bits per 512 bytes of user data is needed. Extending the endurance to 600K cycles requires an even stronger BCH code, since the *raw BER* increases to more than  $10^{-4}$ . Another approach to improve the endurance of the flash memory without increasing the parity information is based on the idea of using the flash as a SIMO channel.

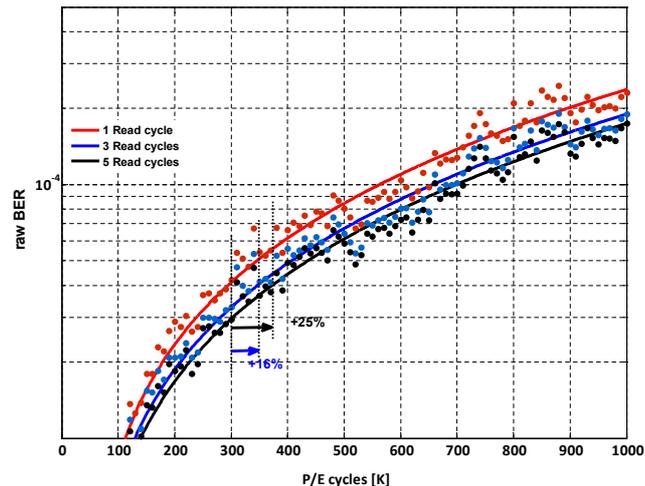


Fig. 2. BER of SLC flash memory.

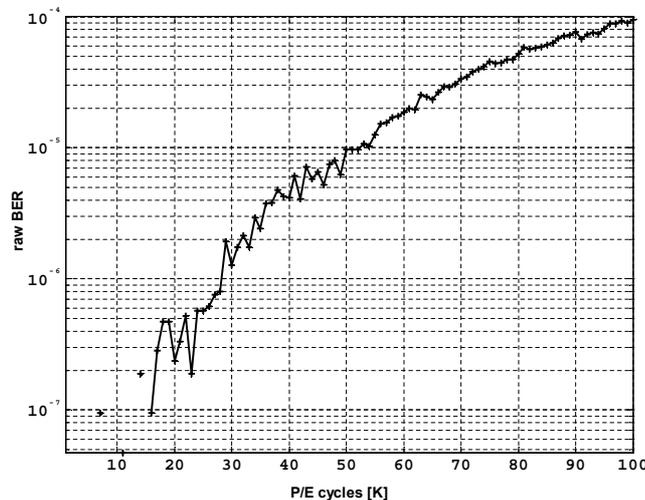


Fig. 3. BER of MLC flash memory.

#### 4. EXTENDING THE FLASH ENDURANCE

In a flash-based storage system that uses BCH codes, the typical procedure followed when a page is read, initially involves the calculation of the BCH syndromes. If the existence of errors is detected, BCH decoding is performed. If the number of errors exceeds the error capability of the code, the data are lost and the device fails.

In the proposed method, when a corrupted flash page can not be recovered by the BCH decoder, a multiple reading process is activated and the data of all pages are compared in order to generate error probabilities for each bit location. Since the SLC channel inserts errors that only change the '1' bits to '0' bits, for a specific bit location, a single bit difference among the pages is enough to mark the respective symbol (bit) as erasure and determine that its correct value is

'1'. It is obvious that as the number of read cycles increases, the method's performance also improves, but with less gain. Even a small number of read cycles can detect a large number of read-related random errors.

Fig. 2 shows the *raw BER* at a block of a SLC memory (red line) and the BER after applying the proposed method with three (blue line) and five (black line) read cycles of each page. As it is shown, the application of the proposed method improves the memory endurance for a few tens thousand P/E cycles. For example, if the SSD uses a BCH which requires a *raw BER* of  $4 \times 10^{-5}$  for guaranteeing the  $10^{-15}$  *user BER*, then the endurance of the device is 300K P/E cycles in a typical configuration, but it increases to 350K P/E cycles (16% improvement) for three read cycles and to 375K P/E cycles (25% improvement) for five read cycles.

## 5. SYSTEM LEVEL ASPECTS

A typical SSD consists of a number of flash channels, a main processor, a host interface and a local DRAM buffer. Each flash channel uses a number of flash chips which share the same bus and a dedicated hardware module, called flash channel controller (FCC), that implements the flash interface (e.g. ONFI), communicates with the main processor and performs error correction, when needed. One or more local busses are used for transferring the data between the main memory, the host interface and the flash channel controllers, using dedicated DMA engines, as shown in Figure 4. At the beginning of the lifetime of a SSD, all blocks are erased, and only read and write commands are executed. when no block erasing is performed, the maximum transfer rate (pages/sec) is determined by the flash channel clock (maximum of 166 Mbytes/sec for ONFI2.0), the bus width and the read/write times.

If  $R$  is the data rate at the flash interface,  $L$  is the number of bytes per flash page and  $T_R$  is the time required for reading a page, then the maximum I/O rate supported during read is  $\frac{R}{L+RT_R}$  [pages/sec]. Typical values for SLC memories are:  $T_R = 25$  usecs,  $L = 4320$  bytes and  $R = 166$  Mbytes/sec. In this case, the maximum supported I/O rate is  $\approx 20$  kIOPs/channel.

Since the time required for accessing the flash cells is comparable or higher than the data transfer time (e.g.  $T_R \approx L/R$ ), one method used for improving even further the maximum data rate per flash channel is pipeline. When pipeline is used, multiple commands are executed simultaneously on different flash chips at the same flash channel, as long as non-overlapping data transfers are guaranteed. When pipeline is used, the maximum supported I/O rate under optimum loading conditions is  $\frac{R}{L}$  [pages/sec]. Using pipeline, the maximum supported I/O rate in the above example is  $\approx 38$  kIOPs/channel.

Nowadays high performance SSDs (HP-SSD) use a large number of flash channels (usually 16), a few Gbytes of SLC

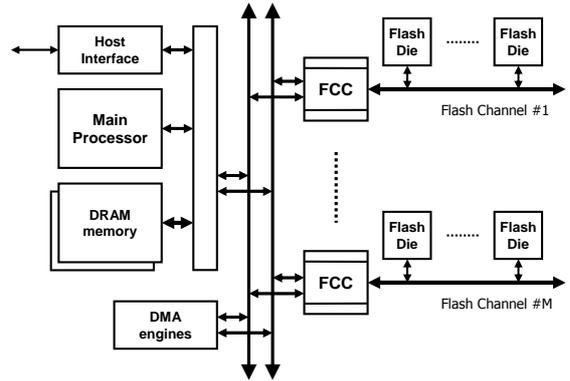


Fig. 4. The Solid-state drive architecture.

memory per channel and their host interface supports data rates of a few Gbps. Based on the above numbers, it would be expected that the maximum supported I/O rate per HP-SSD would be near to the maximum rate supported by the flash channels, 300 kIOPs when no pipeline is used and 600 kIOPs with pipeline. Based on measurements of the most sophisticated HP-SSDs, I/O rates around 120 kIOPs have been reported. That means that the limiting factor of a HP-SSD is not the flash channel but other system parameters, i.e. its internal architecture, the high level functions etc. In this case, using the proposed method for extending the flash endurance, the I/O performance of the flash channel will be decreased, but the I/O performance of the SSD will not be affected seriously.

## 6. CONCLUSIONS

The lifetime of SSDs can be extended by improving the endurance of the used flash memories. This paper proposes a method that exploits the error characteristics of SLC flash memory and by using the flash memory as a SIMO channel and sustains the memory endurance for a few tens of thousands P/E cycles. The application of the proposed method also to MLC memories is under investigation.

## 7. REFERENCES

- [1] A. Marelli R. Michelloni and R. Ravasio, *Error Correction Codes for Non-Volatile Memories*, Springer, 2008.
- [2] Joe E. Drewer and Manzur Gill (Eds), *Nonvolatile Memory Technologies with Emphasis on Flash*, IEEE Press - Wiley Interscience, 2008.
- [3] K. Rose F. Sun and T. Zhang, "On the Use of Strong BCH Codes for Improving Multilevel NAND Flash Memory Storage Capacity," in *IEEE Workshop on Signal Processing Systems (SiPS): Design and Implementation*, Banff, Canada,, Oct. 2006.