A Contribution to the Evaluation of NAND Flash Memory

Jaroslav Kadlec, Radek Kuchta, Radovan Novotný, Zdeňka Kuchtová
Central European Institute of Technology
Brno University of Technology
Brno, Czech Republic
email: jaroslav.kadlec@ceitec.vutbr.cz; radek.kuchta@ceitec.vutbr.cz; novotnyr@feec.vutbr.cz;
xkucht06@stud.feec.vutbr.cz

***Abstract*— NAND flash memories are well known for their uncomplicated structure, low cost, and high capacity. Their typical characteristics include architecture, sequential reading, and high density. NAND flash memory is a non-volatile type of memory and has low power consumption. The erasing of NAND Flash memory is based on a block-wise base. Since cells in a flash chip will fail after a limited number of writes, limited write endurance is a key characteristic of flash memory. There are many noise causes, such as read or program disturbances, retention process, charge leakage, trapping generation, etc. Preferably, all errors in the storage would be adjusted by the ECC algorithm. The conclusion of all mentioned parasitic factors creates a set of external and internal influences which affects variable behavior of memory in time. To prepare an overall analysis of all the important factors that affect the reliability and life-cycle endurance of NAND flash memories and describe the methodology for their evaluation was our main motivation for this paper.**

Keywords- flash memory; non-volatile; bit error rate; error correction code; architecture; reliability.

I. INTRODUCTION

Flash memory has been an important driving force due to the increasing popularity of mobile devices with large storage requirements. Flash memory is respected in many applications as a storage media due to its high access speed, non-volatile type of storage, and low-power consumption. There is a wide range of non-volatile memories, and they all give various characteristics based on the complexity of array organization and structure of the selected cell type [1]. General comparison of NAND and NOR Flash memories is in the Table 1.

Flash memories are becoming widely deployed in many applications, such as solid state drives (SSDs) for embedded controllers and traditional computing storage. NAND Flash memories are becoming more and more popular due to their usage as Solid-State Drives (SSDs) and USB Flash drives which are in general called Flash storage devices.

Another area of application is systems, which allow system reconfiguration, software updates, changing of stored identification codes, or frequent updating of stored information (i.e., smart cards). Electrically erasable and programmable read-only memories (EEPROM's), which are electrically erasable and programmable, will be produced only for specific applications, because they use larger chip areas and are more expensive.

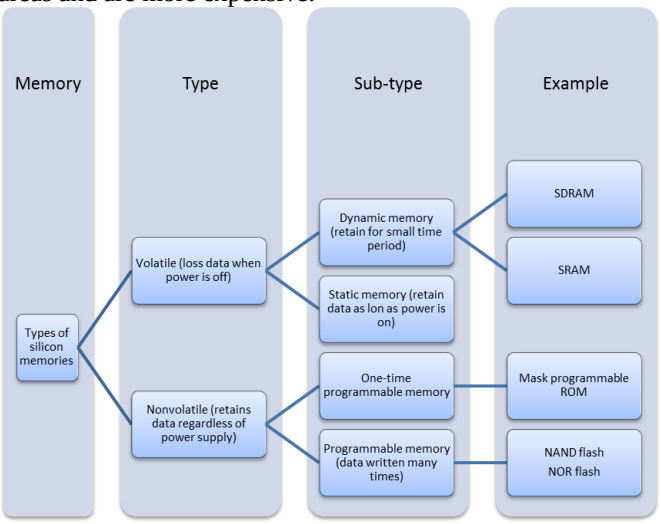

Figure 1. Flash memories as a type of memory device characterized by non-volatility

Following on from these advantages, the manufacturers of memories started to consider the role of flash memories for a new range of applications. These include hard disk caches, solid-state drives, mobile sensor networks, and data-centric computing. Many microcontrollers have integrated flash memory for non-volatile data storage. Flash memory is also used in many applications where data retention in power-off situations and reliability are crucial requirements, such as in embedded computers or wireless communication systems.

Nowadays, flash memory is one of the most popular, reliable, and flexible non-volatile memories to store constant data values and software code. NAND Flash architecture and NOR Flash architecture dominates the non-volatile Flash market [3]. NAND flash requires protracted access time for

 ISBN: 978-1-61208-399-5

the data and subsequent access to any non-consecutive location could be problematic. Nevertheless, once a page of memory is released for read, data can be pulled out from the memory fast, but in general. This is the reason why it was rarely used as the main memory of the system. As a result, there must be a controller to access data which is important in order to manage all the essential tasks of accessing NAND Flash device effectively [2]. General division of the Flash memory types is in the Figure 1.

TABLE I. THE MAJOR DIFFERENCES BETWEEN NAND AND NOR FLASH MEMORY

	NAND	NOR
Memory cell arrangements	Cells are arranged in series with the adjacent cells sharing source and drain.	Cells are arranged in parallel with all the source node of the cells connected to the bit line.
Capacity	tens of Gbits	several Gbits
Non-volatile	Yes	Yes
Interface	I/O interface	Full memory interface
High-speed access	Yes	Yes
Access method	Sequential	Random byte level access
Page mode data access	Yes	No
Performance	Fast read (serial access cycle) Fast write Fasted erase	Fast read (random access) Slow write Slow erase
Price	Low	High
Life Span	10^5-10^6	10^4-10^5
Write cycles	10^6	10^6
Advantages	Fast programing and erasing	Random access, possible byte programing
Disadvantages	Slow random access, difficult byte programing	Slow programing, slow erasing
Typical uses and applications	Storage, file (disk) applications, voice, data, video recorder and any large sequential data archiving	Networking device memory, replacement of EPROM, applications executed directly from non-volatile memory

Due to the non-volatile nature of this storage media, there is a high demand for it in the mobile communication industry. Flash memory has become the most popular choice for mobile devices. NAND Flash memory is commonly found in portable or embedded memory for computers, digital cameras, mobile phones, MP3 players and other devices where data is generally written or read sequentially [4].

II. GENERAL NAND FLASH DEVICE ARCHITECTURE

The overall architecture of the NAND flash device is shown in Figure 2. Unlike most memory technologies, NAND flash is ordered in pages which are written and read as a unit. The elementary unit of operation for a NAND Flash device is one page of data with control commands of the whole block (multiple pages) or the whole chip [2]. Therefore, data can be written only to one page at once. A page is defined as cells linked with the same word line. This is the smallest programmable unit physically made up of a row of cells.

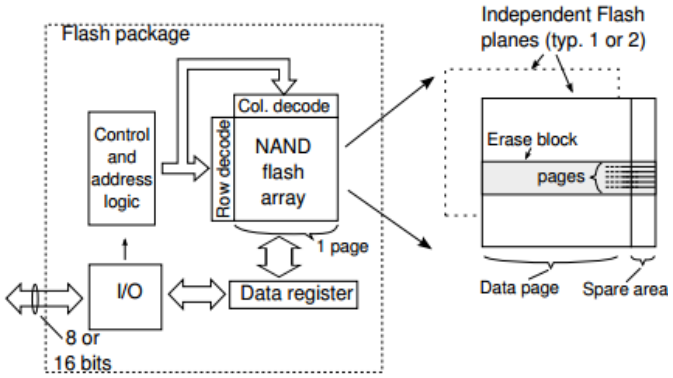

Figure 2. Architecture of the NAND Flash Device [4]

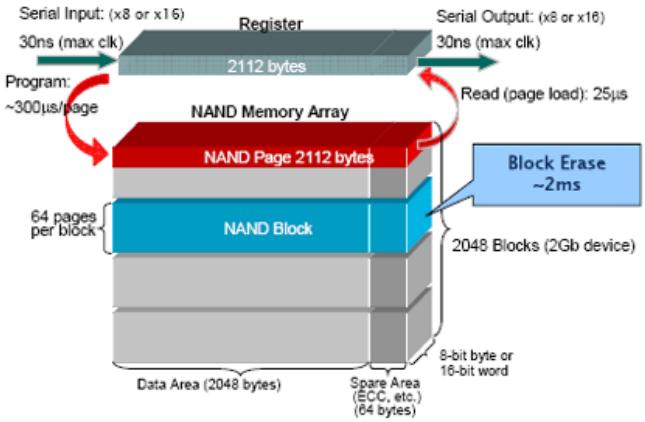

Figure 3. The 2-Gbit NAND device is ordered as 2048 blocks, with 64 pages per block [5].

NAND Flash devices could be considered as large page and small page devices [2]. There are overall 528 bytes (264 words) per small page. For enormous capacities, typically 1 Gbit and more, a large page is used. A large page device usually has 2048 bytes of data and 64 bytes of spare data per page (Figure 3) while a small page device has 512 bytes of data and 16 bytes of spare data per page. The commands sequence for large page and small page devices are different so the controller must be aware of which kind of device is being used.

Cells are organized in pages, and each page is divided into a data area, also named as a “Cell Array” page area, and a redundant area as a spare area for system overhead functions, also named as a “Spare Cell Array” page area. Spare blocks are set apart from the flash storage for remapping bad sectors. This solution prolongs the useful life and reliability of the flash storage device. The spare columns are fully addressable by the user and are typically used for storing Error Correction Code (ECC), wear-leveling, and other organization of information in order to improve data integrity. In operation, bytes from the spare area are equivalent to bytes from the data area and can be used to store the user's data. The spare area is not physically different from the rest of the page.

Before programming, a page must be erased which sets all data bits to “1”. Then, only the value “0” can be

 ISBN: 978-1-61208-399-5

programmed into each cell. An erased, blank page of NAND flash has no charges stored in any of its floating gates. Unlike block-oriented disk drives, nevertheless, pages must be erased in units of erase blocks including multiple pages (typically 32 to 128) before being re-written.

III. ERROR CORRECTION CODE IN NAND FLASH MEMORIES

In digital communication, the quantity of bit errors is the number of received bits of a data stream sent over a communication channel that have been changed due to interference, noise, bit synchronization errors or distortion. The bit error rate or bit error ratio (BER) is the number of bits that have errors divided by the total quantity of transmitted bits throughout a given time interval. BER is a unitless measure, frequently formulated as a percentage. The raw bit error rate relates to the probability of a bit error occurring in an individual bit cell on a flash device [6].

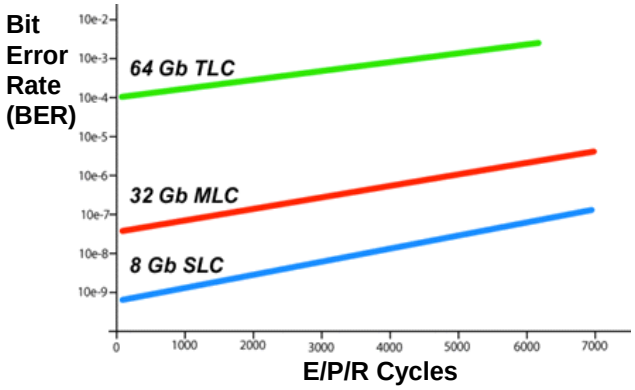

Figure 4. Bit error rate versus Erase/Program/Read cycles for Micron NAND flash [7]

A. *Noise sources in NAND flash and the bit error rate (BER)*

There are many noise causes existing in NAND flash, such as cell-to-cell interference, read or program disturbances, retention process, random-telegraph noise, background-pattern noise, charge leakage and trapping generation, etc. [8]. Such noise sources considerably shrink the storage reliability of flash memory. Over time the quantity of affected cells increases, see Figure 4. Figure 5 shows that Read Disturbances Error Rate is empirically much worse in devices that have consumed erase, program and read cycles than in uncycled devices [7].

Bit errors are a natural consequence of uncertainty when executing any data storage and must be moderated by software or hardware so that the integrity of the original information is not compromised [6]. For NAND flash, this is implemented by using protecting groups of bits with a higher-level error correction algorithm.

Preferably, all errors in the storage would be adjusted by the ECC algorithm. In reality the algorithm protects against a range of errors that are probable to happen.

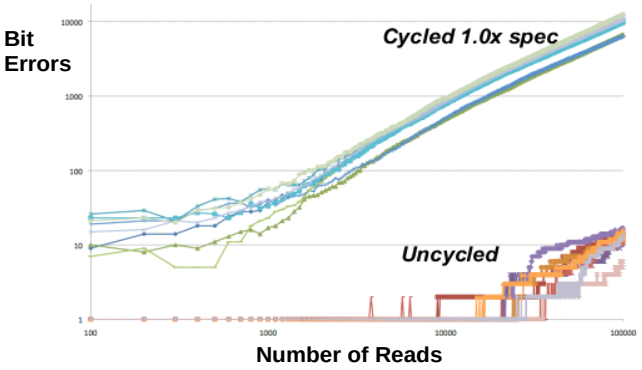

Figure 5. Bit errors versus number of reads [7]

TABLE II. ECC BIT CORRECTION REQUIREMENTS FOR SLC AND MLC NAND FLASH MEMORY [9]

	NAND Process	ECC required	Erase Cycle	Data Retention
SLC	70/60 nm	1-bit	100 K	10 years
	50 nm	1-bit	100 K	10 years
	40/30 nm	4-bit	TBD	10 years
MLC	70/60 nm	4-bit	10 K	10 years
	50 nm	4 ~ 8 bit	5 K ~ 10 K	10 years
	40/30 nm	12 ~ 24 bit or more	3 K ~ 5 K	5 years

Over time NAND flash has augmented storage density by storing more bits per cell and moving to smaller geometries. As NAND Flash memory moves towards more progressive process nodes, the cost of devices is decreasing, but the cells become more vulnerable [10]. The quantity of bits kept per cell is increasing, bit values are represented by smaller voltage ranges, generating more uncertainty in the value stored in the bit cell due to more ambiguity in the amount of charge [6]. As the bit cells get smaller, the individual cells are more vulnerable to failure brought by high-voltage stress because fewer electrons can be trapped in the floating gates. The effect is to narrow the valid voltage ranges for a given value, increasing the probability for program and read disturbances. Since this solution requires higher levels of error correction mechanism in order to ensure the integrity of the data on the flash device, the new technology needs more ECC (Deal, Hamming, RS, BCH, LDPC)[11]. Overview of requirements of ECC for SLC and MLC NAND Flash memory is in the Table 2.

The accepted uncertainty upsurges the probability for data to be stored or read incorrectly, requiring higher levels of error correction for MLC flash than for SLC flash [6]. Devices using NAND flash must integrate very high levels of error correction in order to guarantee support for next generation flash devices – see Figure 6.

 ISBN: 978-1-61208-399-5

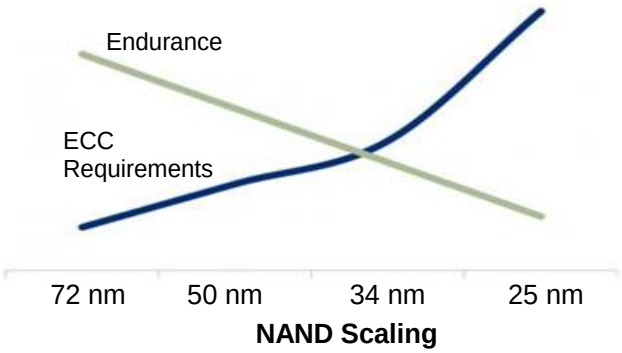

Figure 6. The drawbacks of NAND scaling: decreasing endurance, increasing ECC [12].

A one-bit ECC algorithm is capable of correcting one failure bit per 512 bytes. SLC flash is able to work with single-bit correction over 512 byte sectors because the individual bit error rate is really low.

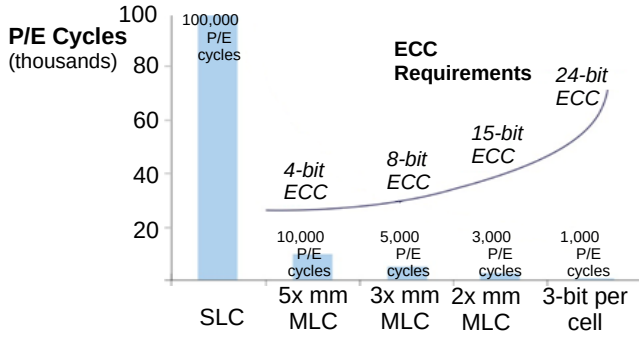

Figure 7. ECC and a life cycle comparison of NAND flash by process node: increase in correction capability is not enough to maintain endurance of the cell [13]

MLC flash has required more powerful correction algorithms capable of correcting four to eight bits to manage the higher bit error rates arising from the greater uncertainty of charging and to detect the various voltage ranges in a single bit cell (see Figure 7) [6].

B. Error detection and correction in NAND Flash Memories

The ECC permits data that is being read or transmitted to be checked for errors and, when necessary, corrected. ECC is a worthy way to recover the incorrect value from the residual good data bits [2]. Error detection and correction or error control includes techniques that permit reliable transfer of digital data by the detection of errors and reconstruction of the original, corrected error-free data. If the ECC cannot correct the error throughout read, it may still detect the error. The application of ECC is used with NAND flash parts to compensate bits that could fail during device operation. On-chip ECC resolves many supposed complications of working with a NAND solution [14]. Currently the error correction is an integral part of the NAND flash that guarantees data integrity.

Up to now, more error correction has been required for MLC NAND technology, whereas SLC NAND has characteristically required only 1-bit ECC for densities up to 4 Gb fabricated at 43 nm [10]. Current trends in the NAND flash market resulting to changes that must be made in the error correction algorithms to preserve the integrity of data stored in next-generation NAND flash devices [6]. The SLC NAND Flash devices, fabricated at 32 nm or 24 nm, require 4-bit or 8-bit ECC, respectively, per 512 bytes [10].

IV. NAND FLASH ECC ALGORITHMS

NAND Flash devices need appropriate error correction algorithms to diminish errors that occur during the programming and read operations [6]. The Life span of NAND Flash could be prolonged without more ECC bits due to the especially proposed operation algorithm. Error detection is usually realized using an appropriate hash function or checksum algorithm. A hash function adds a fixed-length tag to a data, which can be whenever recalculated and verified.

The basic system of ECC theory is to enlarge some redundancy for protection. The redundancy permits the receiver to detect a limited number of errors that may happen anywhere in data, and usually to correct these errors without retransmission. Different ECC techniques are necessary in various types of flash memory.

ECCs are typically divided into two classes: block codes and convolution codes. Hamming codes, Bose-Chaudur-Hocquenghem (BCH) codes, Reed-Solomon (RS) codes, and Low-density parity check (LDPC) codes are most notable block codes and have been widely used in communication, optical, and other systems [8]. The choice of the most effective correction code is a compromise between the number of symbol errors that need to be corrected and the additional storage requests for the generated parity data. Early designs implementing SLC NAND used either no error correction or marginally correcting Hamming codes which offer single error correct and double error detect capabilities [6]. Given the low bit error rates of early flash, this was satisfactory to correct the sporadic bit error that arose. As bit error rates enlarged with each successive generation of both SLC and MLC flash, designers progressed to more complex cyclic codes, such as Reed-Solomon (R/S) or Bose-Chaudhur-Hocquenghem (BCH) algorithms to increase the correction capability [6]. While both of the algorithms are similar, R/S codes execute correction over multi-bit symbols while BCH makes correction over single-bit symbols.

Here is how it works for data storage: when any k-bit data is written to flash memory, an encoder circuit makes the parity bits, adds these parity bits to the k-bit data and creates an n-bit code-word [8]. Parity bits form a code that refers to the bit sequence in the word and is stored along with the unit of data. The routinely computed ECC, i.e., the whole code-word, is kept in the spare area of the page to which it relates. Throughout the reading operation, a decoder circuit examines errors in a code-word, and corrects the mistaken bits within its error capability, thereby recovering the code-word [8].

When the unit of data is demanded for reading, a code for the stored and about-to-be-read word is calculated using the

 ISBN: 978-1-61208-399-5

algorithm. ECC´s are again calculated, and these values are compared to the ECC values held in the spare area. If the codes match, the data is free of errors. The outcome of this assessment yields an ECC "syndrome" that shows whether errors occurred, how many bits are in error, and, if the errors are recoverable, the bit position of incorrect bits. If the codes do not match, the missing or incorrect bits are determined through the code comparison and the bit or bits are corrected or supplied. The additional information represent redundancy added by the code is recycled by the receiver to recover the original data.

A typical ECC will correct a one-bit error in each 2048 bits (256 bytes) using 22 bits of ECC code, or a one-bit error in each 4096 bits (512 bytes) using 24 bits of ECC code. However, as raw BER increases, 2-bit error correction BCH code becomes a desired level of ECC. Next generation flash devices will move to smaller geometries and increased number of bits per cell, features that will increase the underlying bit error rate [6].

V. SUMMARY

Today, flash memory are one of the most popular, reliable, and flexible non-volatile devices to store data. NAND flash memory has become very popular for usage in various applications where a large amount of data has to be stored. This article discusses important aspects related to the storage reliability and the actual bit error rate.

A NAND Flash device is composed by the memory array, which is separated into several blocks. In general it performs three basic operations: program a page, erase a block, and read a page. There are many noise sources that exist in the NAND flash, which considerably shrink the storage reliability of a flash memory. The paper presents a preliminary study, which was conducted in connection with the preparation of an experiment for evaluating the reliability of a NAND flash memory. The purpose of this study was to summarize the theoretical background. The preliminary aim was to identify factors affecting the reliability for potential usage of the methodology of a planned experiment. However, after considering all aspects, it has been realized that this approach is not possible. Therefore, further research will involve life-cycle and reliability testing using the Weibull analysis method.

ACKNOWLEDGMENT

This research has been supported by the European ARTEMIS Industry Association by the project 7H12002 "Interactive Power Devices for Efficiency in Automotive with Increased Reliability and Safety", by the TA04010476 project TACR "Secure Systems for Electronic Services User Verification" and by the CZ.1.05/1.1.00/02.0068 project OP RDI "CEITEC - Central European Institute of Technology".

REFERENCES

[1] Pavan, P., Bez, R., Olivo, P., and Zanoni, "Flash Memory Cells—An Overview," Proceedings of the IEEE, VOL. 85, NO. 8, 1997, pp. 1248-1271.

[2] Eureka Technology Inc. (2012). "NAND Flash FAQ". Retrieved 4. 11. 2014, from http://www.actel.com/ipdocs/apn5_87a_FAQ.pdf.

[3] Gong, B. Y., "Testing Flash Memories", 2004, from http://www.ictest8.com/base/yuanli/Testing%20Flash%20Memories.pdf.

[4] Paikin, A., "Flash memory". Retrieved 4. 11. 2014, from http://www.hitequest.com/Kiss/Flash_terms.htm.

[5] Micheloni, R., Marelli, A., and Commodaro, S., "NAND overview: from memory to systems." Springer Science and Business Media. doi:10.1007/978-90-481-9431-5_2, 2010.

[6] Deal, E., "Trends of NAND Flash Memory Error Correction." Retrieved 7. 8. 2014 from http://www.cyclicdesign.com/index.php/ecc-trends-in-nand-flash, 2009.

[7] Heidecker, J., "NAND Flash Qualification Guideline", NEPP Electronic Technology Workshop, 6. 11. 2012

[8] Wang, X., Dong, G., Pan, L., and Zhou, R., "Error Correction Codes and Signal Processing in Flash Memory", InTech. Retrieved from http://www.intechopen.com/books/flash-memories/error-correction-codes-and-signalprocessing-in-flash-memory, 2011.

[9] Kuo, T.-W., Huang, P.-C., Chang, Y.-H., Ko, C.-L., and Hsueh, C.-W., "An Efficient Fault Detection Algorithm for NAND Flash Memory". Retrieved from http://www.iis.sinica.edu.tw/~johnson/public_files/FaultDetection.pd.

[10] Toshiba Electronics Europe, "How to handle the increasing ECC requirements of the latest NAND Flash memories in your Industrial Design." Retrieved 7. 2. 2013, from http://www.toshiba-components.com/memory/data/Whitepaper_BENAND_11_2012.pdf, 2012.

[11] Deal, E., "Hamming, RS, BCH, LDPC - The Alphabet Soup of NAND ECC." Retrieved 7 2, 2013, from Cyclic Design: http://www.cyclicdesign.com/index.php/parity-bytes/3-nandflash/24-hamming-rs-bch-ldpc-the-alphabet-soup-of-nand-ecc, 2011.

[12] Shimpi, L. A., "Micron's ClearNAND: 25nm + ECC, Combats Increasing Error Rates." Retrieved 7. 2. 2013 from AnandTech: http://www.anandtech.com/show/4043/micron-announces-clearnand-25nm-with-ecc, 2010.

[13] Naftali, S., "Signal processing and the evolution of NAND flash memory." Retrieved 7. 2. 2013, from Anobit: http://embedded-computing.com/articles/signal-evolution-nand-flash-memory, 2010.

 ISBN: 978-1-61208-399-5