

# NAND Flash Memory Organization and Operations

Novotný R\*, Kadlec J and Kuchta R

Brno University of Technology, Technická 3058/10, 616 00 Brno, The Czech Republic

## Abstract

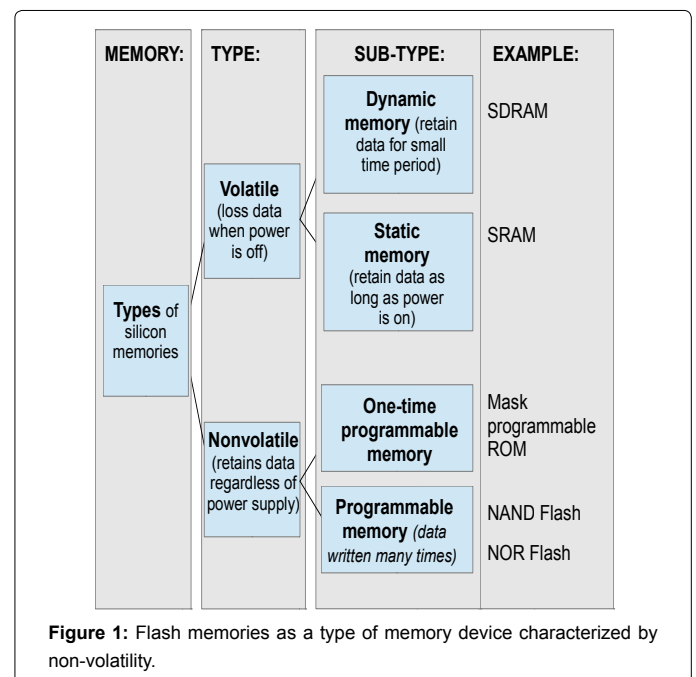
NAND flash memories are well known for their uncomplicated structure, low cost, and high capacity. Their typical characteristics include architecture, sequential reading, and high density. NAND flash memory is a non-volatile type of memory and has low power consumption. The erasing of NAND Flash memory 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 disturbs, 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 behaviour of memory in time. To prepare a review of all the important factors that affect the reliability and life-cycle endurance of NAND flash memories and was our main motivation for this paper.

**Keywords:** Flash memory; Non-volatile; Bit error rate; Error correction code; Architecture; Reliability

## 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 (Figure 1), 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]. 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) [2] and USB Flash drives which are in general called Flash storage devices. Another area of application is as the non-volatile memory in 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 (Figure 2) dominates the non-volatile Flash market [3] because NAND Flash is not byte addressable it is 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 [4]. The major differences between NAND and NOR flash memory (Table 1 and Figure 2). The cell structure of NOR and NAND Memory is shown in Figure 2. The structure of NAND Flash cell is depicted in ( Figure 3). Bold lines in this figure capture current path (with or without wires). Depending on how memory cells are interconnected, it is possible to make a distinction between NAND and

NOR Flash memories. In NAND flash, cells are connected in series, resembling a NAND gate. In NOR flash, cells are connected in parallel to the bit lines. 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 [5].



\*Corresponding author: Novotný R, Brno University of Technology, Technická 3058/10, 616 00 Brno, The Czech Republic, Tel: +420541141111; E-mail: [novotnyr@feec.vutbr.cz](mailto:novotnyr@feec.vutbr.cz).

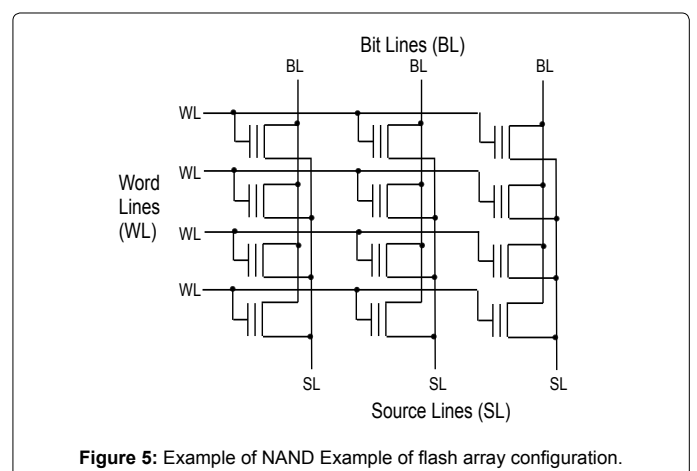
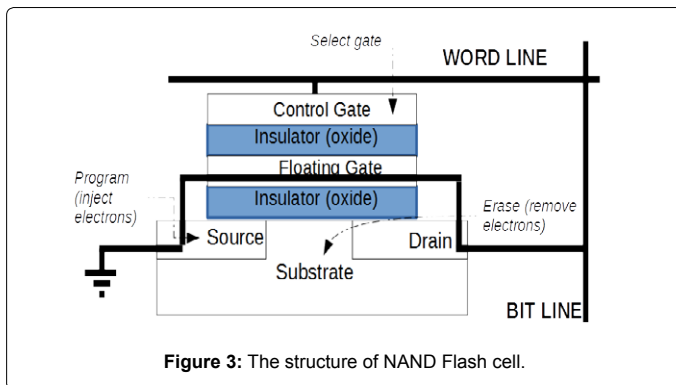
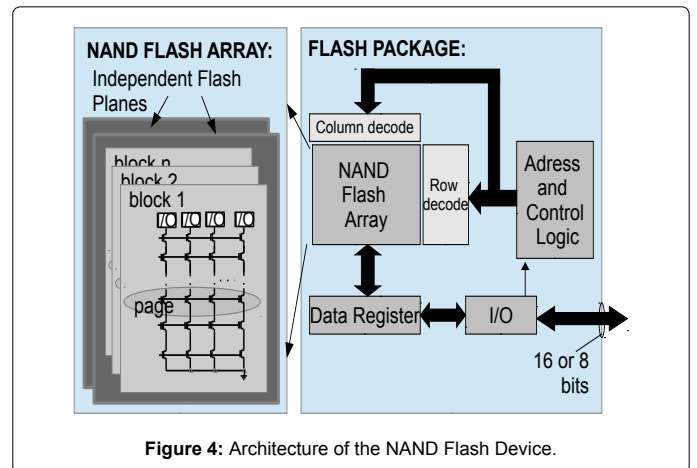
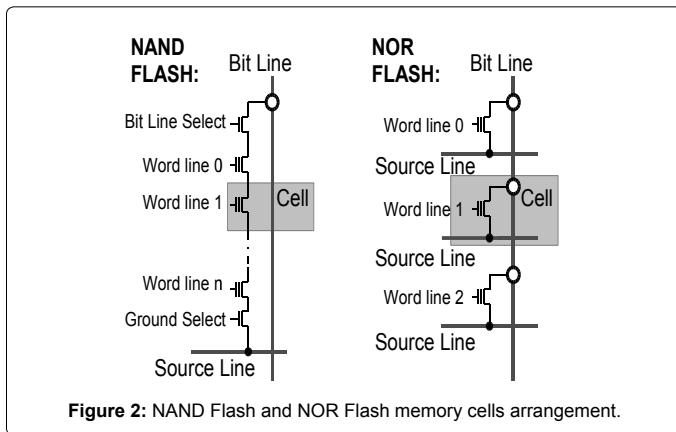
Received August 26, 2014; Accepted January 20, 2015; Published January 30, 2015

Citation: Novotný R, Kadlec J, Kuchta R (2015) NAND Flash Memory Organization and Operations. J Inform Tech Softw Eng 5: 139. doi:10.4172/2165-7866.1000139

Copyright: © 2015 Novotný R, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

	NAND	NOR
Memory cell arrangements (Figure 2)	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, mass data storage.	Several Gbits, code storage.
Non-volatile	Yes, extremely high cell densities.	Yes, larger chip area per cell.
Interface	I/O interface	Full memory interface
High-speed access	Yes, random access.	Yes, serial access.
Access method	Sequential	Random byte level access
Page mode data access	Yes, organized into pages and erased on a block basis.	No, organized on a byte or word basis.
Performance	Fast read (serial access cycle) Fast write Fast erase	Fast read (random access) Slow write Slow erase
Price	Lower cost per bit	Higher cost per bit
Life Span	10 <sup>5</sup> -10 <sup>6</sup>	10 <sup>4</sup> -10 <sup>5</sup>
Write cycles	10 <sup>6</sup>	10 <sup>6</sup>
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

Table 1: The major differences between NAND and NOR flash memory.



### NAND Overview: Flash Array and Architecture

The overall architecture of the NAND flash device is shown in Figure 4. The figure shows the NAND Flash Controller which gives interface to application processors. The page register is a critical data holding area in NAND operations. The register is incorporated in order to receive new data while the data register simultaneously programs the NAND Flash array. 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 [4]. Therefore data can be written only to one page at once. As shown in Figure 5, 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. The page size is typically 2 or 4 Kbytes, while a block contains of 32 or 64 pages [6]. Figure 5 is an example of a flash array configuration. The word line (WL) is the horizontal line

and the correspondent bit line (BL) is the vertical line. When a large number of a floating gate cells need to be operated in the NAND array it is necessary to consider that one floating gate cell is located at every crossing point of word lines and bit lines. Therefore on each NAND string, the bit lines are connected through the string select transistor to a number of memory transistors in series [7] Control gates are linked to the word line, where the decoded address is applied. The source line connects the sources to common grounds, and bit lines connect the drains together and represent data buses. The voltage concurrence applied to the bit line and word line determines an operation-reading, erasing or programming [8]. NAND Flash devices could be considered as large page and small page devices [4]. 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 6) while a small page device has 512 bytes of data and 16 bytes of spare data per page. Main stream devices today have 128/256 pages per block and 8 k/16 k bytes 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. As shown in (Figure 6), 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 (Figure 6). The 2-Gbit NAND device is ordered as 2048 blocks, with 64 pages per block. Each page contains 2112 bytes total, covered of a 2048-byte data area and a 64-byte spare area [9]. Before programming, a page must be erased which sets all data bits to “1”. Then only the value “0” can be 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.

**NAND Flash device: some aspects related to the page, block and plane**

A NAND Flash device is composed by the memory array, which

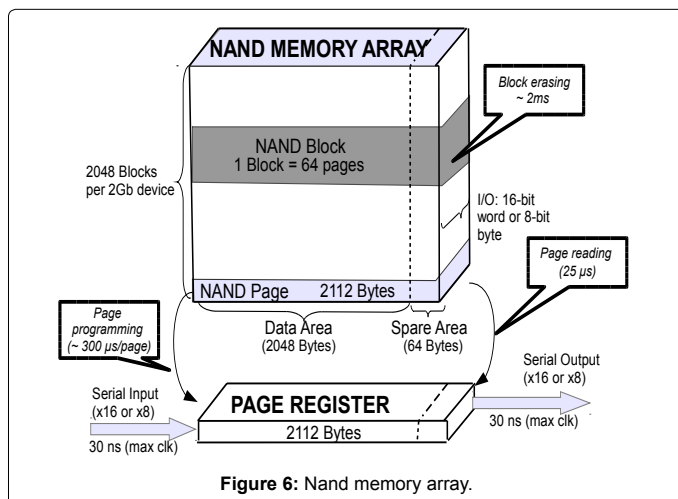


Figure 6: Nand memory array.

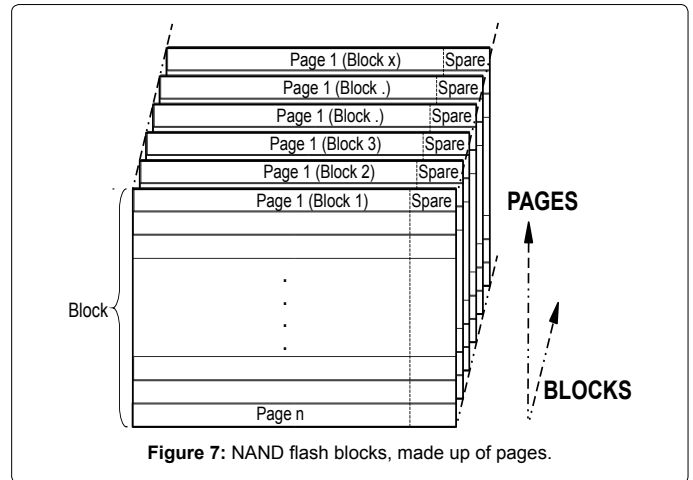


Figure 7: NAND flash blocks, made up of pages.

is separated into several blocks, where a block is the smallest erasable unit of storage (Figure 7) [10]. The reason for this is that all the NAND strings share the same group of word lines that are erased together. Each block involves a set of addressable pages, where MLC typically has 128 pages per block and SLC has 64 pages per block [11]. The basic unit of operation for a NAND Flash device is one page of data with some commands influencing the whole block. However, unlike block oriented disk drives, pages must be erased in units of erase blocks comprising many pages that have previously been re-written. In a block erase operation, a group of consecutive pages is erased in a particular operation. Erase operates on entire blocks and sets all the bits in the block to “1”. Flash devices write in full blocks, which means, that in order to write to a block that may already hold some data the flash controller must move the existing data in the block and combine it with the new data and write all the data back to the flash memory. Read and write are both accomplished in two separate stages, containing the transfer of data over the bus to or from the data register, and the transfer between the data register and the flash array. In order to perform read, program, and erase further circuits are necessary (Figure 7). As the NAND die must be placed in a package with a defined size, it is essential to define a floor plan. The NAND Flash memory is composed of the blocks of pages, which could be grouped into a flash plane. Depending on the kind of device, planes are in principal mutually independent. A single plane covers local buffering for read and program data, and can process operations in parallel. Each plane, involves a set of blocks made up of 64 (SLC) or 128 (MLC) pages. NAND Flash devices can contain independent flash planes, characteristically for storing odd and even blocks, allowing concurrent operations for better performance. On the vertical direction a bit line is highlighted, while a word line is shown in the horizontal direction. All bit lines are linked to the sense amplifiers. The purpose of the sense amplifiers is to transform the current sank of the memory cell to a digital signal. The Row Decoder placed among the planes appropriately bias all the word lines belonging to the particular NAND string. In the peripheral part of the NAND Flash device there are voltage regulators and charge pumps, logic circuits, and redundancy structures [12].

**Flash translation layer, wear leveling and garbage collection**

Flash Translation Layer (FTL) is a software layer, which is between the NAND Flash storage physical media access layer and the File System layer. The FTL is the part of the NAND controller and performs a logical to physical address translation (data locations are represented by their physical addresses), a wear leveling mechanism to prevent the early wear out of block, bad block management, ECC and interleaving

operations. The operating system can write to the memory on a page basis without worrying about the details of its physical address space [7]. Due to the nature of NAND flash, wear out is unavoidable when writing to a NAND flash. A related problem is to write to the same address space and create uneven wear of the memory block. The blocks keeping often-updated data are stressed with a large number of write/erase cycles, while the blocks keeping data updated infrequently are much less stressed. The endurance of the NAND Flash memory can be significantly enhanced taking advantage of wear leveling mechanism which is able to distribute the memory usage uniformly over the memory blocks array [13]. Wear leveling is a process that spreads out the load of frequently rewritten memory blocks over the NAND Flash array as much as possible. When the host application needs an update of the same (logical) sector, the NAND controller dynamically maps the data into a different (physical) sector, keeping track of the mapping. With regard to the need to minimize the impact on performance, garbage collection process is executed in background [7]. There are two types of wear leveling algorithms: dynamic wear leveling and static wear leveling. In case of dynamic wear leveling algorithm the NAND controller internally maintains a map which links the logical block addresses with their corresponding physical Flash memory addresses. The NAND controller then assigns for every write operation a new empty page and links the new page to the original address. In static wear leveling mode, rarely updated static data are kept in "static" blocks while low usage cells are relocated to a new page what results to almost the same number of rewrite cycles [14]. Wear leveling techniques assume that there is the availability of free sectors. When the number of free sectors falls below a critical threshold value, there is an algorithm for sector back-up "compaction" and copying the last valid replication which allows that the obsolete copies can be deleted. Garbage collection prevents erasing a block with valid data and optimizes a process of cleaning the memory. Cleaning policies are set according to criteria used to pick up the block to be erased for garbage collection purposes [7]. NAND Flash memory is free from complex scheduling of the overhead like hard drive disk scheduling, but is affected by the garbage collection issue [15].

## Program and Erase of NAND Memory Array

NAND Flash memory is controlled using set of commands, these sets of commands differ from memory to memory. There are many commands, some are universal to all NAND Flash manufacturers while some commands are manufacture specific and supported only by a few devices. According to the Open NAND Flash Interface (ONFI) Standard there is a list of the basic mandatory command set. The most common commands are 'program', 'read data', 'erase', 'reset', 'program confirmation', 'read status' and 'read ID'. Device specific commands contain 'random read', 'page cache read', 'random write', 'page cache write', 'internal data move', 'two-planes write', 'two planes read', and some others [4]. NAND flash devices carry out three basic operations: program a page, erase a block, and read a page. In a NAND Flash device read and program operations take place on a page basis rather than on a byte or word basis like NOR Flash. This dictates the need to have the size of data I/O register equal to the page size.

### Program a page

NAND Flash devices are programmed on a page by page basis. During the page program operation, a page is written into the data register and then programmed into the memory array. First the page address and the command word are moved into the device followed by the programming data. The programming steps of a NAND Flash device differ depending on whether it is a small block or a large

block device and whether features such as a cache mode or two plane programming are applied. During programming, the ready/busy signal (R/B) is low to indicate that the device is in the busy state. The R/B-signal is low when the NAND chip operates a program, read or erase command and signalizes if the NAND flash device is ready for other operations.

### Erase a block

Flash memory allows only two states: erased and non-erased. A given bit of data can only be written when the media is in an erase state. When data is written into, the bit is considered dirty and unusable for other write operations. A write operation in any type of flash device can only be accomplished on an erased unit and so a write operation must be preceded by an erase operation. In order to must be erased. During era recover the bit to the erase state, a meaningful large block of flash called an erase block, also called as an erase zone, sure operation all cells on the same bit string are erased. In the erase state, a byte can be either all zeroes or all ones depending on the kind of flash device. As a result, the flash technology does not allow the changing of individual bits or bytes from a non-erased state back to an erased state.

### Read a page

Bits in a flash cell are read by changing the voltages on rows and columns of cells followed by assessing the results. In a page read operation, a page is moved from memory into the output data register. NAND Flash devices are read by shifting in the address and command. Once the command and address are shifted in, it requires a few tenths of a micro second to open a page. After a page is opened, data can be lifted out of the device by using the read (RD) command signal. Certain devices support cache read mode and some of them support random data read within a page functionality. NAND Flash memory reads and writes in high-speed, sequential mode, handling data in pages. Differences such as cache programming, random programming and two plane programming are enabled only by some NAND Flash devices.

### Merging log blocks with blocks containing stale data

When units smaller than an erase block are mapped, stale data can remain there. Then the pages affected by the stale data are not accessible for new data. In the limited case of hybrid Flash Translation Layers (FTL), the used process consists of merging log blocks with blocks containing stale data, and programming the result into one or more free blocks [16]. Merge operation can be one of the following three types (full, partial, and switch merge) [17]. A switch merge is applied during sequential writing. The log block contains a sequence of pages precisely replacing an existing data block, and may replace it without any further operation. This means that the old block can then be erased. A partial merge duplicates valid pages from a data block to the log block, after which these two may be substituted. A full merge is desirable when data in the log block is out of order. Valid pages from the log block and the related data block are copied together into a different free block. The original data block and log block are then both erased [5].

### NAND Flash Memory Characteristics

Development of NAND Flash memories has been driven by a gradual progress in novel cell structures and architectural solutions oriented to both reducing cell size and upgrading product functions. The NAND flash memory has become an indispensable component in embedded systems for its flexible features. Applications may need lesser or greater erase counts, different error correction capabilities, and a range of storage longevity requirements. NAND flash devices



differs in many parameters and characteristics that include cell types, architectural, performance, timing parameters and command set. The following chapter provides an overview of the typical characteristics.

### Architecture, Sequentially reading, and high density

NAND Flash architecture based on independent blocks was introduced by Toshiba in 1989. In the NAND-flash, unit cells are linked sequentially, where the cells resemble a NAND gate. This layout and architecture inhibits cells from being read and programmed independently. These cells must be read in series. Due to their excellent scalability and performances, NAND Flash has achieved very high density in terms of bits per mm<sup>2</sup> and feature size scaling.

### Flash memory non-volatility and low power consumption

Flash memory is non-volatile, which implies that it retains data even without being powered-of. Non-volatility comes from the types of transistors used which are floating gate transistors. Since stored data stays even when the memory device is not electrically powered it does not need power to maintain its data. NAND flash is the only memory, which offers both GB density and non-volatility.

### NAND Flash memory programming and erasing based on block-wise base

Flash memory has the ability to be programmed and erased electrically, hence combining the advantages of EPROMs and EEPROMs. They increased flexibility compared to electrically programmable read only memories (EPROM's), which are electrically programmable but erasable via ultraviolet (UV) radiation. The dissimilarity between the Flash Memory and EEPROM lies in the fact, that EEPROM erases and rewrites its content byte by byte. Since flash memory erases or writes its data in entire blocks, this makes it very fast in comparison with EEPROM. A single cell could be electrically programmable and a large amount of cells (block, sector, or page) are electrically erasable almost at the same time. Since NAND flash does not offer a random-access external address bus the required data are read on a block-wise basis (also termed as page access), where each block keeps hundreds to thousands of bits, similar to a kind of sequential data access.

### Serial storage properties

From the system designer's perspective, the biggest difference is that the NAND Flash is a serial storage device whereas most other memories are random access memory (RAM). Serial storage device requires longer access times for obtaining data. The effect is that NAND flash as a serial storage device does not give a random-access external address bus and needs a special NAND Flash controller to access data and therefore is hardly ever used as the main memory of the system. System designers must reflect these differences when interfacing the end system with the NAND Flash device. Since most microprocessors and microcontrollers require byte-level random access this is one of the main explanations why NAND-flash is inappropriate to replace the RAM memory. The speed at which the DRAM or SRAM may access data and also their capability to address at byte level is incomparable with the Flash memory.

### Noise in the reading process

In the NAND flash the only way to access the discrete cell for either reading or writing is across the other cells in its bit line. This enhances noise related to the read process [18], and also needs attention during writing to ensure that adjacent cells in the string are not disturbed. Many of the more difficult characteristics of NAND flash are due to this

organization, which removes much of the decoding overhead found in other memory technologies.

### Restricted write endurance

Since cells in a flash chip will fail after a limited number of writes limited write endurance is a key characteristic of flash memory [19]. Infrequently cycled blocks will have longer retention and often cycled blocks will have shorter retention. It is highly important to employ a wear leveling mechanism, which ensures equal memory block load rather than cycling and potentially destroying the same block. Wear leveling is necessary on MLC (multi-level cell) devices where blocks can normally support less than 10,000 erase program cycles and offers additional advantages on SLC (single-level cell) devices where blocks can offer up to 100,000 erase-program cycles.

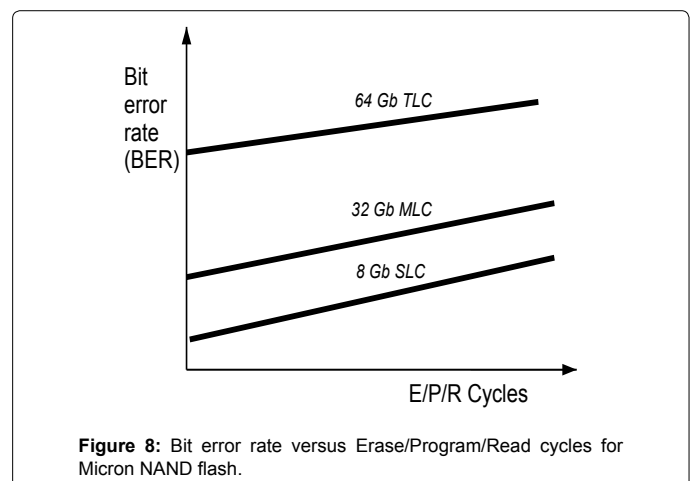
### Some extended features

There are some extended features offered in addition to those basic ones:

- Device operation status read and manufacturer ID read,
- One-time-programmable area (OTP area) to keep vendor unique data such as serial number,
- Locking or unlocking the blocks to avert data loss on unintended software performance,
- Internal movement of the block into another memory location escaping time consuming data relocations from and back to chip (copy-back),
- Only a portion of a page may be programmed at a time and the rest may be programmed at some other time, avoiding block erasing (partial page programming),
- Boot-like feature, page 0 is loaded into data register automatically after reset or power-on,
- Supplementary cache register (cache operation) for read operation or pipe-lined program (Figure 8).

### 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 unit less 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 [20]. Figure 8 shows that the bit error rate (BER) is much worse in parts that have consumed erase, program, read cycles, and is different for SLC, MLC and TLC NAND Flash technology.

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

There are many noise causes existing in NAND flash, such as random noise, cell-to-cell interference (inter-cell interference), read or write disturb, programming errors, retention process (retention errors), random-telegraph noise, background-pattern noise, charge leakage and trapping generation, etc. [21]. Such noise sources considerably shrink the storage reliability of flash memory. Over time the quantity of affected cells increases (Figure 9). This figure shows that Read Disturb Error Rate is empirically much worse in devices that have consumed erase, program and read cycles than in uncycled devices [22]. Quality of data retention for uncycled or cycled devices is a natural consequence of the memory cell limited life (Figure 9). 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 [20]. For NAND flash, this is implemented by using protecting groups of bits with a higher-level error correction algorithm. To reduce possible errors, Error Correcting Codes (ECC) are widely used in NAND memories. Through ECC it is possible to fill the discrepancy between the error probability offered by the memory and the desired error probability. ECC algorithm takes care of the failures during the life of the device and improves the reliability of the read operation in the customer final application [7]. Preferably, all errors in the storage would be adjusted [23]. In reality the algorithm protects against a range of errors that are probable to happen (Figure 10). The drawbacks of NAND scaling: decreasing endurance, increasing ECC [24]. 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 [25]. The quantity of bits kept per cell is increasing, and 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 [20]. 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

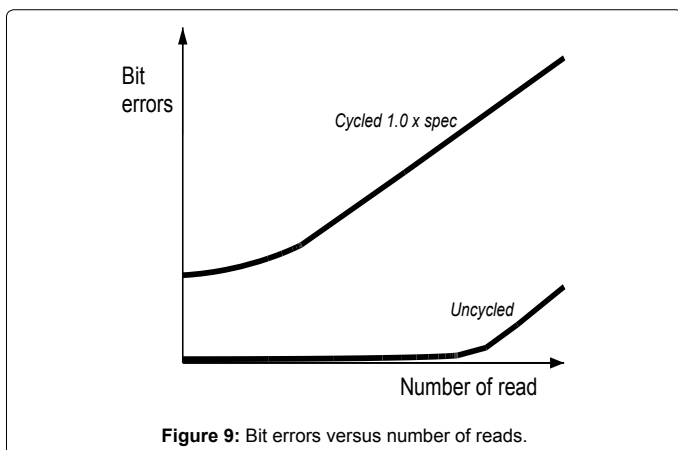


Figure 9: Bit errors versus number of reads.

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

Table 2: Error correction for MLC flash and for SLC flash.

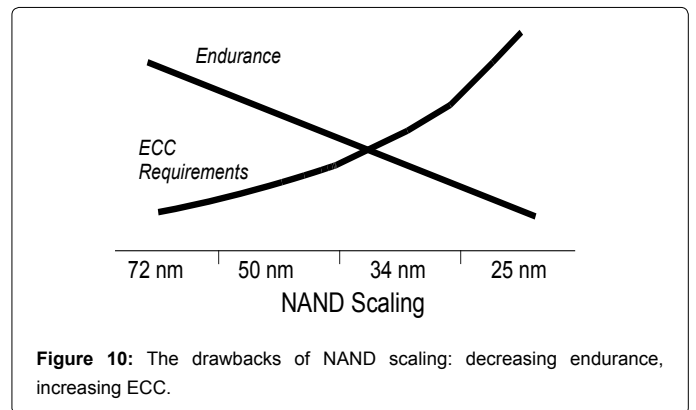


Figure 10: The drawbacks of NAND scaling: decreasing endurance, increasing ECC.

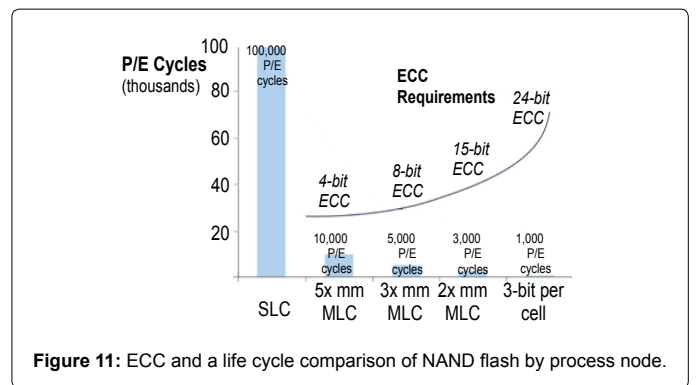


Figure 11: ECC and a life cycle comparison of NAND flash by process node.

the integrity of the data on the flash device, the new technology needs more Error Correction Code (Deal, Hamming, RS, BCH, LDPC) [26]. 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 [20] (Table 2). Devices using NAND flash must integrate very high levels of error correction in order to guarantee support for next generation flash devices (Figure 10). 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 (Table 2) [24].

Multi-Level Cell (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 (Figure 11) [20]. For example the industry has started to deploy LDPC (low-density parity-check) in SSD controllers in order to improve error correction capability [27] (Figure 11). 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 [28].

**Error detection and correction in NAND Flash Memories**

The error correction code (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 [4]. 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 error correction code resolves many supposed complications of working with a NAND solution [29]. On chip ECC techniques have rarely been adopted [30], NAND controller includes a hardware that supports ECC calculations. 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 Gbits fabricated at 43 nm [25]. 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 [20]. The SLC NAND Flash devices, fabricated at 32 nm or 24 nm, require 4-bit or 8-bit ECC, respectively, per 512 bytes [25].

### NAND Flash ECC Algorithms

NAND Flash devices need appropriate error correction algorithms to diminish errors that occur during the programming and read operations [20]. 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 (Table 3). 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. Error correction codes are typically divided into two classes: block codes and convolution codes. The difference between these codes is the encoding principle [31]. Block coding works with messages of fixed length. In the block codes, the information bits are followed by the parity bits.

NAND Type	SLC	MLC	TLC
<b>Datasheet ECC Requirement</b>	1-bit ECC per 528 bytes of data	12-bit ECC per 539 bytes of data	60-bit ECC per 1146 bytes of data
<b>Bit Error Rate</b>	0.02%	0.28%	0.65%
<b>Suggested ECC</b>	SEC/DEC Hamming Code or Reed-Solomon Code	BCH Algorithms	Low Density Parity Check (LDPC) Codes

Table 3: ECC recommendations.

Error Correction Level	Bits Required in the NAND Flash Spare Area		
	Hamming	Reed-Solomon	BCH
1	13	18	13
2	N/A	36	26
3	N/A	54	39
4	N/A	72	52
5	N/A	90	65
6	N/A	108	78
7	N/A	126	91
8	N/A	144	104
9	N/A	162	117
10	N/A	180	130

Table 4: Number of bits required for various ECC correction strengths.

In convolutional codes the information bits are spread along the sequence [32] Hamming codes, Bose-Chaudur-Hocquenghem (BCH) codes [33], 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 [22]. 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 [20]. 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-Chaudhuri-Hocquenghem (BCH) algorithms to increase the correction capability [20]. While both of the algorithms are similar, R/S codes execute correction over multi-bit symbols while BCH makes correction over single-bit symbols (Table 4). Number of bits required for various ECC correction strengths [34]. 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 [22] 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 [22]. When the unit of data is demanded for reading, a code for the stored and about-to-be-read word is calculated using the 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. The decoding phase can reduce read performance as well as the memory response time. 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 [20].

### Summary

Today, flash memory is 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 NAND Flash memory, 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 NAND flash, which considerably shrink the storage reliability of flash memory [35]. The paper presents a preliminary technology review, which was conducted in connection with the preparation of an experiment for evaluating the reliability of

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 statistical planned experiment (DOE, Design of Experiments) [36]. 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 [37].

#### Acknowledgement

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" 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, Zanoni E (1997) Flash Memory Cells—An Overview. In Proceedings of the IEEE 85: 1248-1271.
2. Agrawal N, Prabhakaran V, Wobber T, Davis J, Manasse M, et al. (2008) Design Tradeoffs for SSD Performance. In USENIX Technical Conference, Boston, USA.
3. Gong Yi (2004) Testing Flash Memories. Available online.
4. NAND Flash FAQ (2013)
5. Boboila S, Desnoyers P (2010) Write endurance in flash drives: measurements and analysis. In Proceedings of the 8th USENIX conference on File and storage technologies, San Jose, USA 115-128.
6. Choi GS, Mankyu S (2014) Investigating Page Sizes in NAND Flash Memory. Proceedings 12: 377-379.
7. Micheloni R, Crippa L, Alessia M (2010) Inside NAND Flash Memories. Springer: New York, USA.
8. Paikin A (2004) Flash memory. Hitequest.com.
9. Nand Flash (2013) <http://www.cheadledaterecovery.co.uk/wpcontent/uploads/2013/01/Nand.gif>
10. Sandvido M, Chu F, Kulkarni A (2008) NAND Flash Memory and Its Role in Storage Architectures.
11. Cooke J (2014) The Inconvenient Truths of NAND Flash Memory.
12. Micheloni R, Marelli A, Commodaro S (2010) NAND overview: from memory to systems. Springer: Netherlands.
13. Campardo G, Tiziani F, Liculo M (2011) Memory Mass Storage. Springer: Heidelberg, Dordrecht, London, New York.
14. Joshi Y, Kumar P (2012) Energy Efficient Thermal Management of Data Center. Springer: New York, Dordrecht, Heidelberg, London.
15. Lim SH, Choi H, Kyo P (2007) Journal Remap-Based FTL for Journaling File System with Flash Memory. In High Performance Computing and Communications: Third International Conference, Houston, USA, HPCC, 192-203.
16. Gal E, Toledo S (2005) Mapping Structures for Flash Memories: Techniques and Open Problems. In Software-Science, Technology and Engineering. Proceedings. IEEE International Conference 83-92.
17. Gupta A, Kim Y, Urganonkar B (2009) DFTL: A Flash Translation Layer Employing Demand-based Selective Caching of Page-level Address Mappings. In 14th international conference on Architectural support for programming languages and operating systems (ASPLOS'09), Washington, DC, USA, 229-240.
18. Aritome S, Shirota R, Hemink G, Endoh T, Masuoka F (1993) Reliability Issues of Flash Memory Cells. In Proceedings of the IEEE 81: 776 - 788.
19. Desnoyers P (2009) Empirical Evaluation of NAND Flash Memory Performance. In ACM SIGOPS Operating Systems Review 44: 50-54.
20. Deal E (2013) Trends of NAND Flash Memory Error Correction. Cyclic Design
21. Wang X, Dong G, Pan L, Zhou R (2013) Error Correction Codes and Signal Processing in Flash Memory.
22. Heidecker J (2012) NAND Flash Qualification Guideline. In 3rd NASA Electronic Parts and Packaging (NEPP) Program Electronic Technology Workshop USA.
23. Yang C, Emre Y, Chakrabarti C (2012) Product Code Schemes for Error Correction in MLC NAND Flash Memories. In Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 20: 2302-2314.
24. How to handle the increasing ECC requirements of the latest NAND Flash memories in your Industrial Design (2013)
25. Deal E (2011) Hamming, RS, BCH LDPC-The Alphabet Soup of NAND ECC
26. Kuo W, Huang C, Chang H, Ko L, Hsueh W (2014) An Efficient Fault Detection Algorithm for NAND Flash Memory.
27. Shimpi LA (2010) Micron's Clear NAND: 25nm + ECC, Combats Increasing Error Rates.
28. Naftali S (2013) Signal processing and the evolution of NAND flash memory.
29. Serial NAND Flash Memory Flyer(2011) micron.com
30. Micheloni R, Marelli A, Ravasio R (2014) Error Correction Codes for Non-Volatile Memories. Springer: Netherlands.
31. Tomashevich V, Hanus P (2014) Convolutional Codes.
32. Micheloni R, Marelli A, Eshghi K (2012) Inside Solid State Drives (SSDs). Springer: London.
33. Carlo S, Fabiano M, Indaco M, Prinetto P (2013) Design and Optimization of Adaptable BCH Codes for NAND Flash Memories. In Microprocessors and Microsystems 37: 407-419.
34. Micron Technology (2013) NAND Flash 101: An Introduction to NAND Flash and How to Design It In to Your Next Product.
35. Grupp L, Caulfield A, Coburn J, Swanson S, Yaakobi E, et al. (2009) In Characterizing Flash Memory: Anomalies, Observations, and Applications. MICRO'09, New York, USA.
36. Montgomery D (2012) Design and Analysis of Experiments, 8th Edition. John Wiley and Sons: New York.
37. Zhao K, Zhao W, Sun H, Zhang T, Zhang X (2013) LDPC-in SSD: Making Advanced Error Correction Codes Work Effectively in Solid State Drives.