

MLC NAND Flash memory: Aging effect and chip/channel emulation[☆]



Antonios Prodromakis, Stelios Korkotsides, Theodore Antonakopoulos^{*}

Department of Electrical and Computer Engineering, University of Patras, Patras 26504, Greece

ARTICLE INFO

Article history:

Received 29 December 2014

Revised 30 April 2015

Accepted 10 June 2015

Available online 27 June 2015

Keywords:

Non-volatile memories

NAND Flash

Memory aging

FPGA emulator

ABSTRACT

This work presents an FPGA-based emulator that can be used for emulating NAND Flash memories, either at the chip or at the channel level, along with the effect of aging on their performance. The emulator is based on a reconfigurable hardware-software architecture, which enables accurate representation of various NAND Flash technologies, focusing especially on MLC cases. The presented architecture can be used for emulating memories at the chip and channel level, while the proposed hardware platform can be used as a valuable tool for developing and evaluating memory-related algorithms and techniques. In this paper, we analyze the architecture of the NAND Flash memory emulator and we present details about its internal functionality. Using experimental results, we demonstrate the high accuracy achieved when it is used to emulate specific MLC and TLC NAND Flash chips and we describe how this custom hardware can be used to emulate a complete NAND Flash channel, which consists of multiple NAND Flash chips that share a common data path and support the execution of pipelined commands.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

NAND Flash memories, the most well established non-volatile memory (NVM) technology today, are used extensively for replacing magnetic hard disk drives and volatile memory-based caches in enterprise storage systems. NAND Flash-based solid state drives (SSDs) proliferate as a low-cost, high-performance and reliable storage medium for commercial and enterprise storage systems. The rapid scaling of NAND Flash memories, with process nodes down to 15 nm, and the use of multi-level cell (MLC) and triple-level cell (TLC) technologies has increased their storage density and reduced the storage cost per bit dramatically. However, their lifetime capacity has also been affected. Different noise sources and interferences along with aging effects have a great impact on the memory reliability and endurance, and hence, on the storage systems where these memories are used. Numerous methods and techniques, such as wear-leveling, specialized error correcting codes (ECC) and precoding techniques have been employed to compensate these effects [1–4], while other techniques, more complex but also more efficient, like dynamic adaptation of read reference thresholds, are at an experimental level [5].

The development of these techniques is based on experimental characterization of NVM cells and chips. Characterization is related with measuring bit error ratio (BER) and response times (read and write times) during the whole lifetime of a device, for various loading data patterns and timing scenarios. This process is performed using real NVM ICs, usually the engineering or pre-production parts, while more thorough testing at the system level is performed when production parts are available. This approach has two major drawbacks. On one hand, it is a very time-consuming process, since the aging of a NVM may require a large number of program/erase (P/E) cycles to be performed for each experiment, ranging from tens of thousands (NAND Flash) to millions (Phase Change Memory, PCM) program cycles. On the other hand, the aging characteristics of a NVM are proportionally dependent on the number of the performed P/E cycles, thus, making it impossible to conduct different or successive experiments at the same aging state of a memory chip. In [6] we presented a model that accurately represents the aging process of a MLC NAND Flash cell, while in [7] the analysis of a MLC NAND Flash memory as a time-variant communications channel, based on the asymmetric 4-PAM model, was presented. In [8] we presented the architecture of a flexible FPGA-based platform, designed for accurate emulations of NVM technologies at the chip level.

For achieving high storage capacity, minimum latency and high I/O rates, SSDs use multiple NAND Flash memory channels that have distinct data paths and operate asynchronously. In this case, a number of NAND Flash chips is used per channel by using the same data/control signals and a small number of different

[☆] This is an extended version of the work "A Versatile Emulator for the Aging Effect of Non-Volatile Memories: The case of NAND Flash" presented at the 17th Euromicro Conference on Digital System Design, 2014.

^{*} Corresponding author.

E-mail addresses: aprodromakis@upatras.gr (A. Prodromakis), stelkork@ece.upatras.gr (S. Korkotsides), antonako@upatras.gr (T. Antonakopoulos).

control/status signals. Such architecture allows the use of pipelining, in order to increase the channel write performance by exploiting the much higher write time compared to the data transfer time. In this paper, we present the architecture of a flexible FPGA-based platform, designed for accurate emulation either of a single NAND Flash chip or of a NAND Flash channel that supports pipelining, focusing mainly on high-density MLC and TLC technologies. Accuracy is measured in reference to experimentally specified bit error probabilities for various aging conditions (ie. the number of P/E cycles applied to a NAND Flash chip), usually for random data patterns. The hardware platform presented in this work is based on a reconfigurable hardware–software architecture which enables the accurate emulation of any NAND Flash technology. The initial designs were implemented on a FPGA with a hard-wired CPU core for higher flexibility, while more complex configurations of the proposed platform were implemented on a high-end FPGA. This hardware platform is a valuable tool for evaluating memory-related algorithms, signal processing and coding techniques.

The remainder of this paper is organized as follows. Section 2 analyzes the basic characteristics, the most common architectures and I/O interfaces of NAND Flash memories. Section 3 analyzes the level distributions of two NAND Flash memory technologies and how they are affected by the aging conditions. In Section 4 we discuss the key aspects of emulating the BER of a NAND Flash cell, while in Section 5 we present the main architectural components of a more complex system, where the NVM emulator has been configured to perform emulation at a chip or at a channel level. Finally, in Section 6 we present experimental results of NAND Flash chips and demonstrate how the proposed system accurately emulates their BER behavior.

2. NAND Flash memory technology

NAND Flash memories are based on Flash cells, implemented by using floating gate transistors (FGTs), that is, field effect transistors (FETs) with an additional floating gate between the substrate and the control gate, as depicted in Fig. 1(a). The effective threshold voltage of a FGT, and thus its I - V characteristic, depends on the charge stored in its floating gate (Fig. 1(b) for SLC NAND) [9]. NAND Flash cells acquire non-volatile properties as the floating gate is surrounded by dielectrics which ensure the reliable isolation of the trapped charge for long periods of time [10]. Fig. 1(c) illustrates the interconnection of FGTs in a NAND Flash block. Groups of FGTs sharing the same bit-line are connected in series, forming strings, while logical pages are formed by cells sharing the same word-line. All strings of cells sharing the same group of word-lines form a NAND Flash block. There are several architectures which determine how pages are formed within word-lines. For example, in the odd/even bit-line architecture of a NAND Flash, odd pages are formed by cells belonging to the odd bit-lines, while even pages are formed by cells belonging to the even ones, respectively. However, in the all bit-line (ABL) architecture there is no such separation [11]. Without loss of generality, in this work we assume that each word-line of cells contains only one logical page.

Programming of NAND Flash cells is achieved by applying bias voltages to the control gate and the drain of the FGTs, which causes the Fowler–Nordheim (FN) tunnelling phenomenon and traps the charge into the floating gate. This operation is only allowed if the cell was previously in the erased state (no charge stored in the floating gate). Information stored in the cells is read by applying a small voltage at the drain of the FGT and sensing the current that flows through it. Write (usually referred as program) and read commands are performed on a page basis, while erase is performed on a block basis.

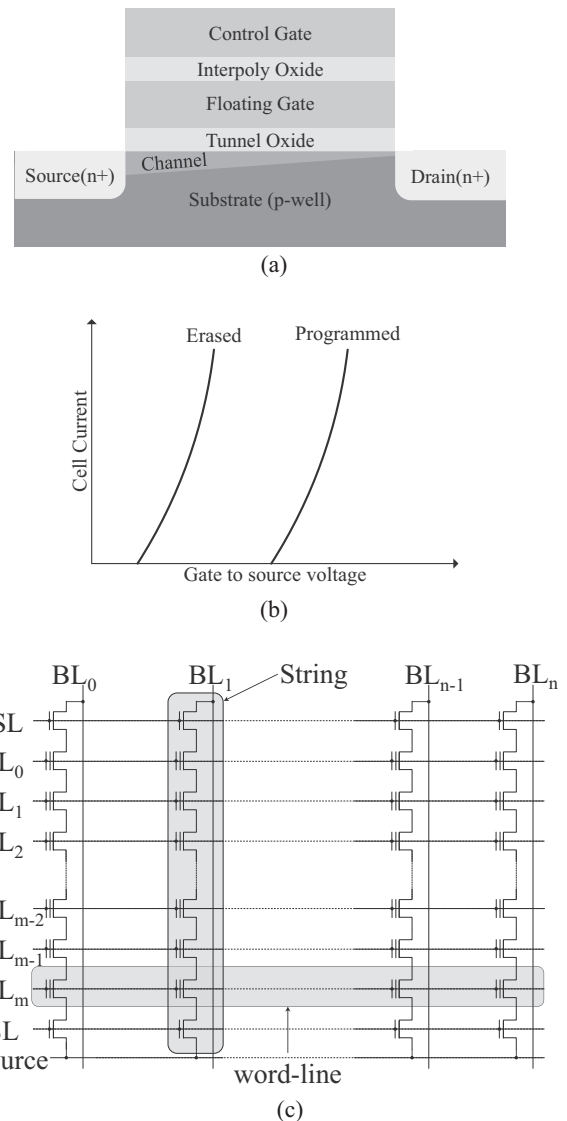


Fig. 1. (a) Floating gate transistor, (b) I - V characteristic and (c) NAND Flash block.

Fig. 2(a) illustrates a simplified block diagram of a NAND Flash chip, which consists of the 2D memory cell array, cache and data buffers, program and read circuits, registers, control logic modules and the physical layer to interconnect with the NAND Flash controller. The most common NAND Flash I/O interfaces take advantage of control signals (CE, CLE, ALE, WE, RE) which define the operations to be performed. Additionally, a bidirectional 8-bit bus (DQ) is used for command, address and data transfers and a Ready/Busy signal (R/B) is used to indicate the target status. The response time of a NAND Flash device is determined by the data transfer time across the DQ bus and the access time of the cell array, which is approximately 20–90 μ s for read (t_{rd}) and 300–2500 μ s for program (t_{pg}) operations. As technology scales and page size increases, the relation between the data transfer time and the read/write times plays an important role on the sustained data rate that can be achieved. Hence, for decreasing the data transfer time, the latest NAND Flash I/O interfaces use double data rate (DDR) asynchronous logic, as well as a data strobe signal (DQS) to achieve high data rates. The most common I/O interfaces of NAND Flash chips are Toggle [11] and ONFI [12], with data rates up

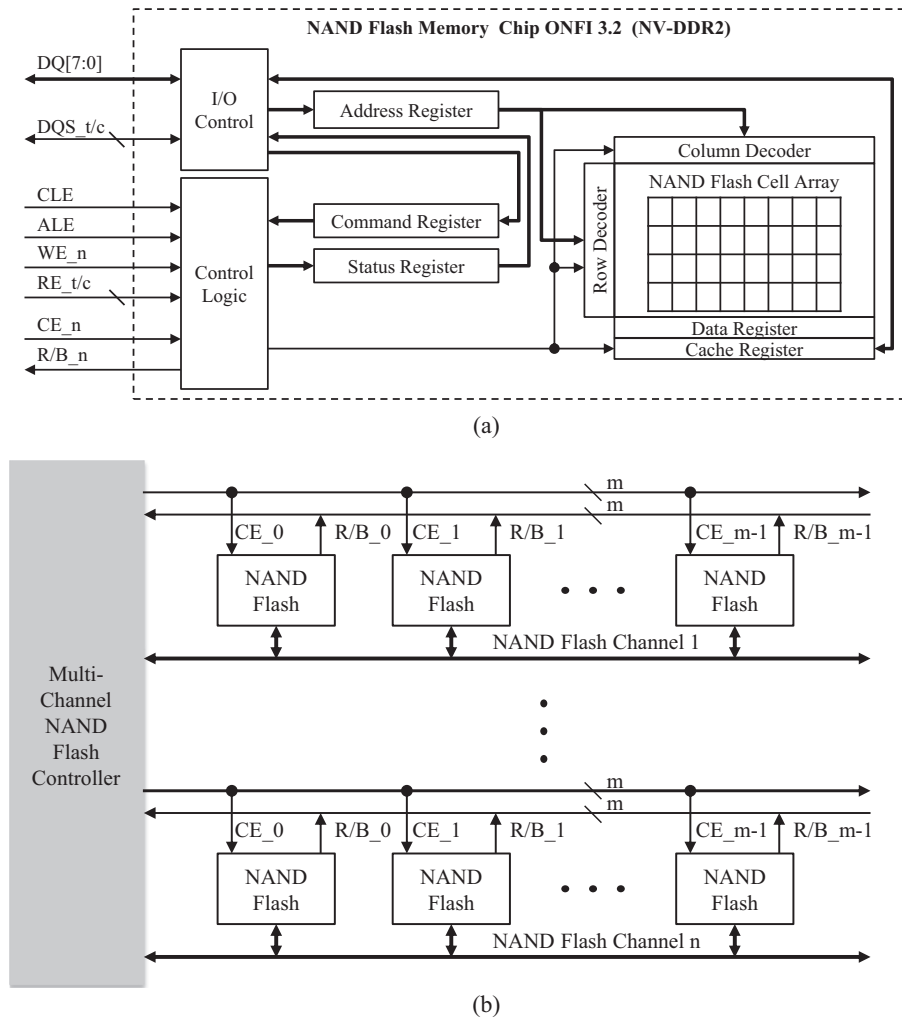


Fig. 2. (a) Block diagram of a NAND Flash device. (b) Multi-channel architecture with multiple targets per channel.

to 400 MBps. The I/O signals used on the NV-DDR2 mode of ONFI3.2 are illustrated in Fig. 2(a). Usually the read time (the time required to read the data from the memory cells and to store them in the chip's internal buffer) is comparable to the data transfer time, while the write time (the time required to store the data from the chip's internal buffer into the memory cells) is a multiple of the data transfer time, and that results in lower sustained data rate during write compared to read.

In order to increase the storage density and to implement high performance NAND Flash SSDs, a number of NAND Flash chips is connected to the SSD controller using a set of independent channels. Each channel uses a number of Flash chips that share the same data path and a subset of control signals, while a few dedicated control/status signals are used per Flash chip. Each Flash memory has independent CE and R/B signals and thus operations can be performed on an interleaved manner, maximizing the utilization of the channel, allowing pipeline execution of consecutive read/write commands at different dies. The SSD controller accesses multiple channels in parallel in order to increase the overall system throughput. A multi-channel architecture of a NAND Flash based system with multiple Flash chips per channel is depicted in Fig. 2(b). Using this approach, the read sustained data rate may achieve the maximum channel data rate under optimum loading conditions, while the write sustained data rate improves significantly, usually by a factor equal to the pipeline depth.

3. Modeling the effect of aging

3.1. Modeling level distributions

Storing data in a MLC NAND Flash is achieved by accurately programming its cells into various intermediate voltages. More specifically, for an n -bit/cell NAND Flash, each cell can be programmed into 2^n different levels. Each level corresponds to a symbol, represented as an n -bit binary vector, which can be mapped using different schemes (i.e. Gray mapping, direct mapping) [1]. As the number of states increases, the margin separating them becomes less, therefore MLC memories are more vulnerable to noise sources than SLCs. The voltage/level distributions are affected by different noise sources, such as cell-wearing, while cell-to-cell interference (CCI) plays a major role when process nodes below 30 nm are used [13]. It has been shown in [14,15] that each NAND Flash cell can be modeled as a level-dependent additive white noise channel (LD-AWGN).

The noise characteristics depend on the aging state of the cell and the input symbol s , i.e. the noise is data-dependent. Let L_s denote the ideal level of the input symbol s . The read-out signal S is a random variable given by (1), where μ_s and σ_s are the mean and standard deviation of the LD-AWGN:

$$p(S) = \frac{1}{\sqrt{2\sigma_s^2\pi}} e^{-(S-(\mu_s+L_s))^2/2\sigma_s^2} \quad (1)$$

Cycling a memory cell alters the parameters μ and σ of the Gaussian probability density function. The mean noise levels are usually shifted to higher values and level distributions become wider. Consequently, level distributions of different symbols may increase their overlap and an erroneous read of the stored information is more likely to happen, therefore leading to higher raw BER.

3.2. Multi-level and triple-level cell technologies

In this paper, we use the results of [16] to determine the mean level L of each symbol in a MLC NAND Flash cell. In addition, we extend the model presented in [7] to cover TLC NAND Flash devices as well.

Fig. 3(a) and (b) present the level distributions of a MLC and a TLC NAND Flash cell, respectively. We assume that if σ denotes the AWGN standard deviation of an intermediate level then the outer levels (erased and fully programmed states) have a standard deviation of $k_1\sigma$ and $k_2\sigma$, respectively. This approach can cover all types of MLC and TLC NAND Flash technologies presented in the

existing literature. For example, based on [17], $k_1 = 2$ and $k_2 = 1$, while based on [18], $k_1 = 1.5$ and $k_2 = 1.2$. Additionally, $k_1 = 4$ and $k_2 = 2$ as in [1,2].

Furthermore, we have parameterized the nominal voltage levels in order to easily adapt the mathematical analysis to the electrical characteristics of different devices.

The nominal voltage levels of a 4-levels MLC NAND Flash can be expressed as:

$$\begin{aligned} L_1 &= \alpha W \\ L_2 &= (\alpha + m_1)W \\ L_3 &= (\alpha + m_1 + 1)W \\ L_4 &= (\alpha + m_1 + m_2 + 1)W \end{aligned} \quad (2)$$

In a TLC NAND Flash memory the nominal voltage levels are:

$$L_i = \begin{cases} \alpha W & \text{when } i = 1 \\ (\alpha + m_1 + i - 2)W & \text{when } 2 \leq i \leq 7 \\ (\alpha + m_1 + m_2 + 5)W & \text{when } i = 8 \end{cases} \quad (3)$$

In order to minimize the effect of erroneous charge detections, which usually happen between adjacent voltage distributions, we assign Gray code mappings to the symbols in both MLC and TLC technologies.

By treating the memory device as an asymmetric n-PAM communication channel with time-variant (aging) characteristics (the case of 4-PAM has been analyzed in [7]), one can express the relation of BER with μ and σ of the Gaussian probability density function in a closed form. Furthermore, this analysis can be extended for non-equiprobable data and for different noise models determined by k_1 and k_2 . Using the same methodology, we can treat the TLC NAND Flash cell as an asymmetric 8-PAM communication channel with time-variant (aging) characteristics.

The probability $P(e_s|s_i)$ of symbol error when reading a symbol s_i , can be computed by integrating each voltage distribution for $S \notin [T_{i-1}, T_i]$. Therefore, $P(e_s|s_1)$ is given by:

$$P(e_s|s_1) = \frac{1}{2} \operatorname{erfc} \left(\frac{T_1 - L_1}{k_1 \sigma \sqrt{2}} \right) \quad (4)$$

Similarly, we can compute the probabilities of symbol error for all distributions with standard deviations equal to σ . Therefore, for $i \in [2, 7]$:

$$P(e_s|s_i) = \frac{1}{2} \operatorname{erfc} \left(\frac{|T_{i-1} - L_i|}{\sigma \sqrt{2}} \right) + \frac{1}{2} \operatorname{erfc} \left(\frac{T_i - L_i}{\sigma \sqrt{2}} \right) \quad (5)$$

Finally, the probability of symbol error for s_8 is given by:

$$P(e_s|s_8) = \frac{1}{2} \operatorname{erfc} \left(\frac{T_7 - L_8}{k_2 \sigma \sqrt{2}} \right) \quad (6)$$

The total symbol error probability is equal to symbol error rate (SER) and is a function of symbol probabilities (p_i), mean values ($\mu_i + L_i$) and standard deviations (σ_i) of the threshold voltage distributions:

$$SER = P(e_s) = \sum_{i=1}^8 p_i P(e_s|s_i) \quad (7)$$

In a TLC NAND Flash, BER calculation depends on symbol mapping. If we consider the case of Gray mapping, which is the most commonly used, we can make the approximation that since two adjacent symbols differ only in one bit, the probability of bit error $P(e_b|s_i)$ when receiving a symbol s_i is approximately:

$$P(e_b|s_i) \approx \frac{1}{3} P(e_s|s_i) \quad , i \in [1, 8] \quad (8)$$

and BER is equal to the total probability of bit error:

$$BER = P(e_b) = \sum_{i=1}^8 p_i P(e_b|s_i) \quad (9)$$

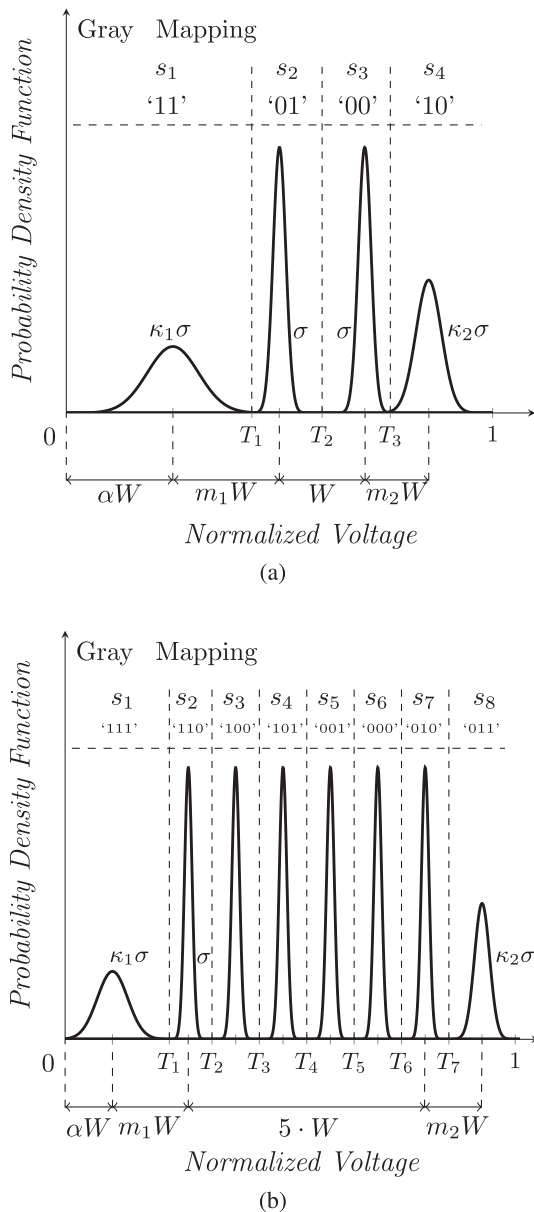


Fig. 3. Level distributions of (a) 2-bits/cell MLC NAND Flash and (b) 3-bits/cell TLC NAND Flash.

Table 1
Characteristics of different MLC NAND Flash devices.

	Device A	Device B	Device C	Device D
Number of blocks	8,192	4,096	16,384	16,384
Pages per block	128	64	128	128
Page size (Bytes)	4096 + 128	2048 + 64	4096 + 224	8192 + 448
Total capacity (Gbits)	32	8	64	128
Page read time (μ s)	60	25	25	35
Page program time (μ s)	800	200	230	300
Block erase time (ms)	2.5	2	0.7	0.7

4. Emulating the BER of a NAND Flash cell

4.1. Relationship between P/E cycles and noise characteristics

In order to emulate the aging behavior of a NAND Flash cell, the relationship between the aging state, that is the number of P/E cycles experienced by the memory cells and the noise characteristics has to be determined. However, noise characteristics, that is mean and standard deviation of the distributions, are statistical metrics which cannot be directly measured in a typical memory device. In this section, we introduce a methodology of computing the aging-noise relationship.

The qualitative similarity between BER as a function of σ and BER as a function of P/E cycles indicates that there is a relation between σ and the number of P/E cycles. In [14], it is stated, with a justification based on measurements, that this is a linear relation during the nominal lifetime of an MLC NAND Flash memory device, $\sigma = aPE + b$. We have verified this by studying measurements from several MLC memories and different statistical characteristics of the Gaussian distributions. Table 1 presents the characteristics of the MLC devices used in our experiments. Moreover, as shown in Fig. 4a, the linear relationship is preserved in all three different MLC NAND Flash models and more specifically, for the first model ($k_1 = 4, k_2 = 2$) $a = 8.48 \cdot 10^{-5}, b = 0.01345$, for the second model ($k_1 = 4, k_2 = 1$) $a = 9.57 \cdot 10^{-5}, b = 0.01347$ and for the third model ($k_1 = 1, k_2 = 1$) $a = 11.69 \cdot 10^{-5}, b = 0.01329$.

On the other hand, by using the measurements of [19], we observe that the relationship between σ and the number of P/E cycles in a TLC device is not linear, but it can be expressed as a second order polynomial equation, following the relationship $\sigma = cPE^2 + dPE + e$. The coefficients in the case of the first model are $c = -4.126 \cdot 10^{-11}, d = 1.059 \cdot 10^{-6}$ and $e = 0.01898$, for the second model $c = -4.259 \cdot 10^{-11}, d = 1.109 \cdot 10^{-6}$ and $e = 0.01933$, while for the third model $c = -4.199 \cdot 10^{-11}, d = 1.142 \cdot 10^{-6}$ and $e = 0.01958$.

The importance of this observation lies on the fact that the bit error emulation of an NVM can be accomplished with high precision by measuring its aging behavior, without any knowledge of its internal architecture or the electrical specifications of its cells. However, if we are interested in emulating the internal electrical characteristics of memory cells (e.g. the threshold voltages in a NAND Flash memory cell), then the mean values of the distributions must be provided, since they cannot be acquired by mere observations.

4.2. Memory cell emulation

As analyzed in Section 3, the BER behavior of a NAND Flash cell can be modeled as a level dependent AWGN communication channel (LD-AWGN), with time-variant characteristics due to aging. The architecture for emulating such a cell, depicted in Fig. 5b, consists of three modules. The Noise Mapping module takes as input the mean and standard deviation values of the $k = 2^n$ distributions,

as well as their nominal levels, and stores them to internal RAM modules.

The noise characteristics can be provided either in terms of P/E cycles or directly as 32-bit single precision μ, σ parameters. In the former case the Aging Logic block maps the user-specified aging condition (P/E cycles) to equivalent noise characteristics (σ) based on the provided distribution model and the analysis of Section 4.1. Although this module can be implemented on hardware, a software implementation in the embedded processor is preferable, since it provides the flexibility to emulate different technologies using different distribution models by the same hardware setup.

The Noise Mapping module selects the noise characteristics which correspond to the input symbol s and provides them to the LD-AWGN module. The latter is responsible for generating the soft read-back signal S which represents the actual read voltage if the symbol s was written to the memory at the specified P/E cycles. Finally, the Hard Decision module implements the decoding of the read-back signal to an n -bit symbol s' . The hard-decision is taken based on the provided read reference thresholds or by

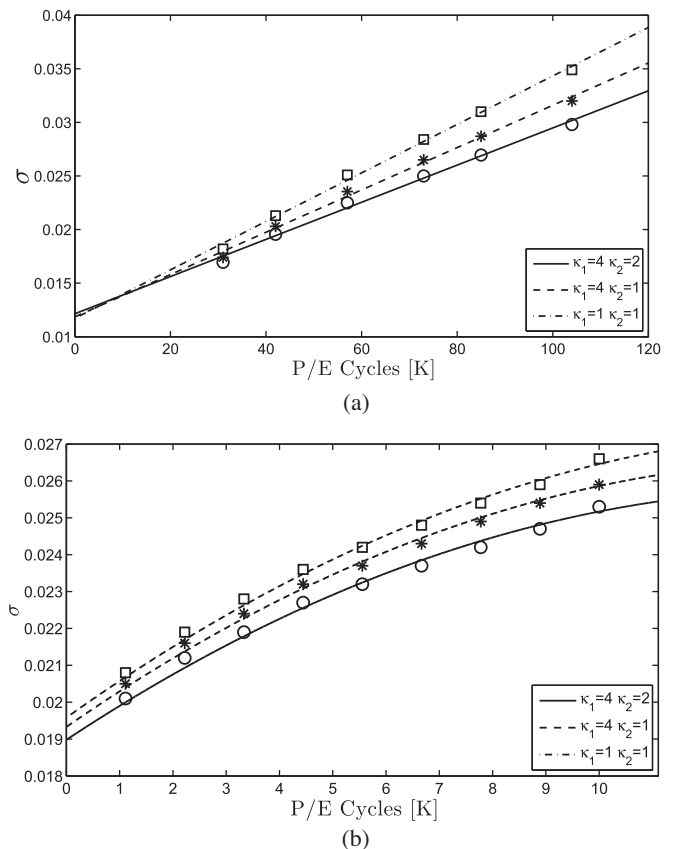


Fig. 4. Standard deviation as a function of P/E cycles for different models of NAND Flash (a) 2-bits/cell MLC NAND Flash and (b) 3-bits/cell TLC NAND Flash.

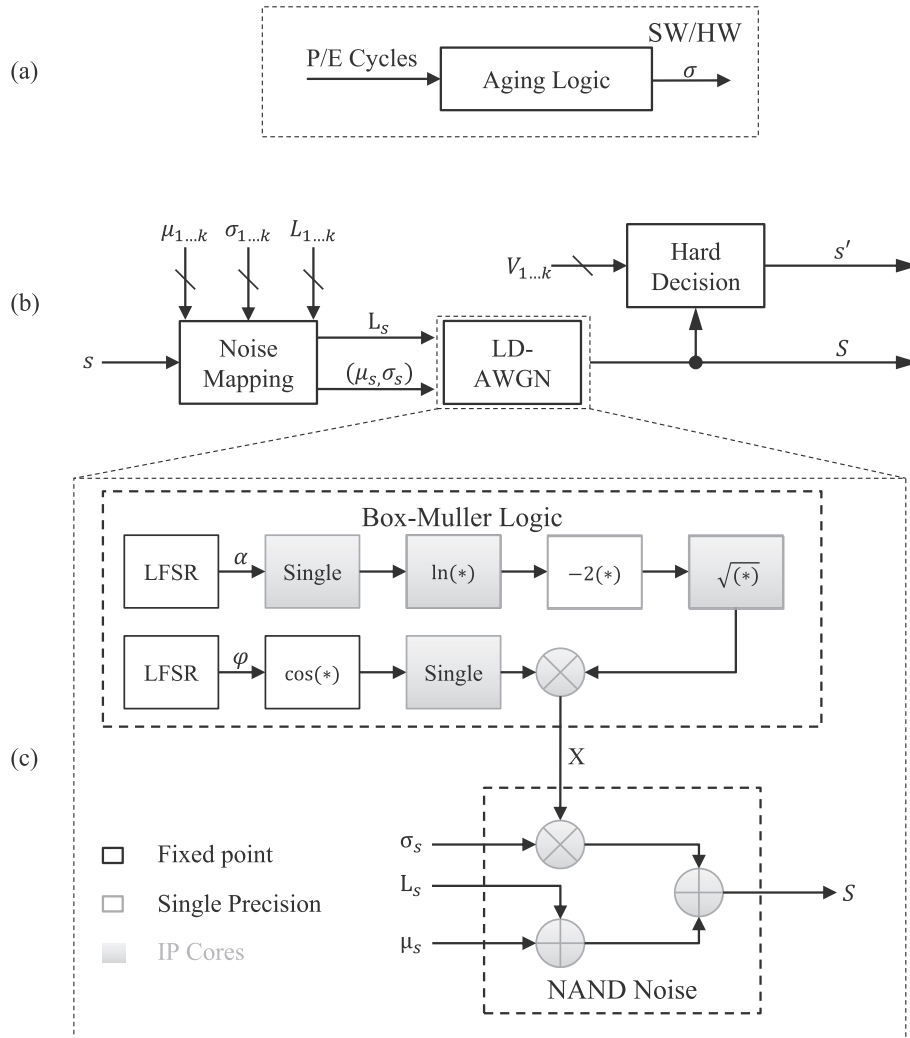


Fig. 5. Emulation logic block diagrams: (a) Aging Logic, (b) communications channel and (c) LD-AWGN.

applying a dynamic adaptation of read reference thresholds algorithm. The proposed architecture offers high flexibility, since it can support emulation of cells with different storage capabilities (SLC, MLC, TLC, QLC), by adjusting on-the-fly the value of k .

4.3. LD-AWGN implementation

The implementation of the level dependent AWGN generator, shown in Fig. 5c, is based on the Box-Muller method [20]. According to this method, if a and φ are independent random variables from the same uniform density function on the intervals $(0, 1)$ and $(0, 2\pi)$ respectively, and

$$X = \sqrt{-2 \cdot \ln(a)} \cdot \cos(\varphi) \quad (10)$$

then X will be a variable from the normal distribution with unit variance, and zero mean ($X \sim \mathcal{N}(0, 1)$). Finally, if

$$S = (X \cdot \sigma_s) + (\mu_s + L_s), \quad (11)$$

then $S \sim \mathcal{N}(\mu_s + L_s, \sigma_s^2)$.

The implementation of the LD-AWGN module, is based on the architecture described in [6]. In our current implementation, the LD-AWGN module operates at 200 MHz, providing a read-back signal S per clock, that is a processing rate of 50 MBps for the MLC technology. Depending on the available resources, a larger number

of LD-AWGN modules operating in parallel can be implemented, increasing the total processing rate.

5. Architecture of the NAND Flash emulator

5.1. Emulating a NAND Flash chip

The aim of the proposed design is to develop a high performance emulation platform, capable of interfacing with existing NAND Flash controllers and emulating the behavior (in terms of BER characteristics and response times) of different NAND Flash technologies accurately. Emulation can be performed at any user-specified state of the aging process, eliminating the need for cycling real NAND Flash chips. Furthermore, it provides the capability to perform different experiments at the same aging conditions, which is not possible when real chips are used.

In [8] we analyzed a system configuration, where the I/O interface of a general purpose NVM-Emulator (NVM-E I/O) has been specified for interfacing with a microprocessor, using the AMBA AXI4 specifications. This approach provides the advantage of performing BER experiments, for different NAND Flash technologies, using the same hardware set-up. The general term NVM-E is used since the same approach can be used not only for NAND Flash emulation, but also for other memory technologies, like PCM. Since the emulation of the BER characteristics of a NAND Flash

can be performed independently to its I/O interface, these experiments can be executed at higher processing and transferring rates than those of the real system. This is mainly due to the ability of eliminating the t_{pg} time during emulation. For instance, when 32 LD-AWGN modules are used in parallel, the processing rate is four times faster than currently used controllers, and two times faster than the emerging ones (ONFI4-based controllers).

In this work we describe the implementation of a NAND Flash emulator that is compatible with a typical ONFI interface, the ONFI3.2 NV-DDR2 interface. However, this analysis can be easily extended to other types of interfaces, like Toggle. Fig. 6 highlights the architecture of the NVM-E core, the block diagram of which has been configured at a module-level abstraction in order to demonstrate its structural similarity with a real NAND Flash chip. The advantage of this approach is that a NAND Flash chip can be replaced by the NAND Flash emulator transparently. Hence, the development of various signal processing algorithms and techniques can be applied directly to the data acquired by the NAND Flash emulator in the same way as if a real chip were used. The NAND Flash core consists of the following modules:

The *Control Logic* and the *I/O Control* modules implement the logic to interface with the NAND Flash controller. The former handles the I/O control signals and contains the finite state machines, which are activated depending on the command to be executed. The latter is responsible for handling the data bus I/Os. Commands, addresses and data are processed via this module and are stored in internal registers and buffers.

The *Command* and *Status Registers* have the same functionality with the typical registers of a NAND Flash device. The *NVM-E Registers* are used to store the aging parameters, such as the mean and standard deviation values, the ideal voltage levels and the read reference thresholds, as well as other parameters related to the emulation target, like the cell technology and the page/block sizes. Finally, the *Timing Registers* are used to store the response (program, read and erase) times of the chip. Their content can be

adjusted according to the aging conditions, based on the experimental results of real NAND Flash chips.

The *Data In Buffers* are used to store the input data temporarily. When the data of a word-line has been loaded to the buffers, the symbols are then stored to a DDR3 DRAM module. The DRAM DIMM interfaces with the rest of the emulator logic using a 64-bits (DQ) DDR3 interface. When the symbols have been processed by the emulation logic during an ONFI read command, they are stored as output page data to the *Data Out Buffers* and as soft word-line data to the *Soft Out Buffer*. The read and write addresses are acquired by the *Address Decoder*, which translates the row/column NAND Flash addresses to the flat address space of the DRAM.

5.2. Emulation-specific commands

The NAND Flash emulator supports all the basic NAND Flash commands, like Page Program, Page Read, Block Erase, Set/Get Features, Reset and Read Status commands. In order to provide full compatibility with the existing NAND Flash controllers, the access of the additional registers of the NVM-E core is performed using custom Set/Get features commands. Furthermore, the soft data can be read from the NAND Flash core by applying a Soft Read command when a Page Read command has been executed. The Soft Read command has been implemented similarly with the Page Read command, differing only to the command opcode and page size.

5.3. Synthesis results

The NVM-E core has been also synthesized for the Xilinx's xc7vx690tffg1157-2 Virtex-7 FPGA device using the Vivado 2014.4 suite. Table 2 presents the post-synthesis utilization of various configurations of the NVM-E core (the number of LD-AWGN cells increases). Furthermore, it illustrates the processing rates that

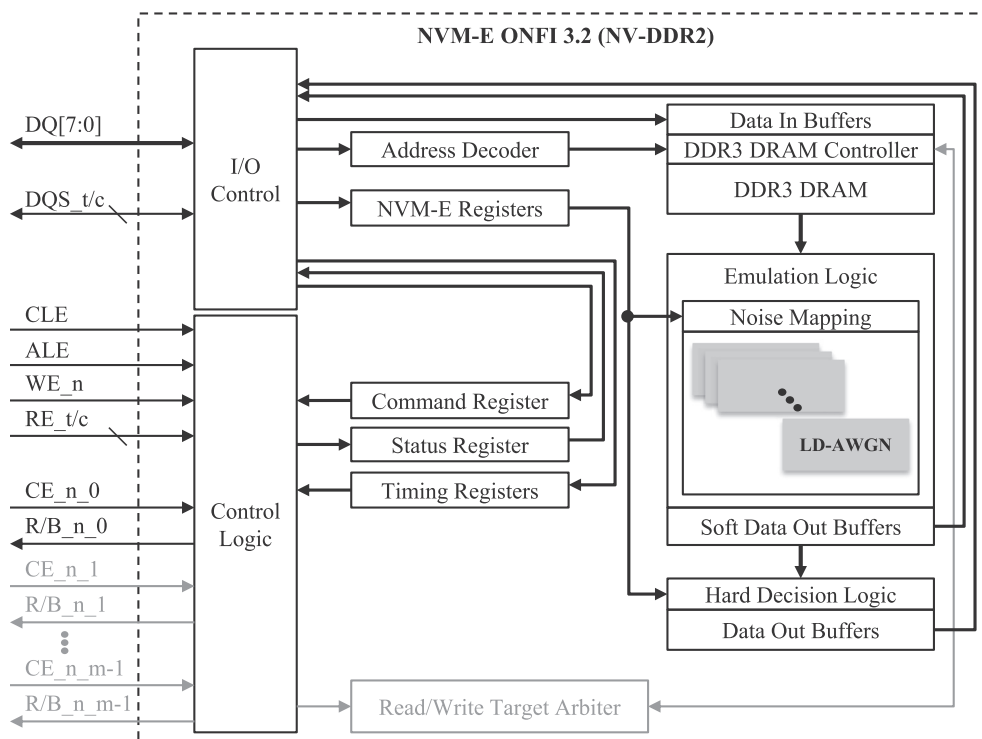


Fig. 6. Architecture of the NVM-E.

Table 2
Synthesis results.

	4 Cells	16 Cells	32 Cells	64 Cells	Available
Slice LUT	10,192	39,363	79,582	157,205	433,200
Slice registers	19,032	68,248	134,251	266,768	866,400
BRAM	15	36	64	134	1,470
DSP	48	192	384	768	3,600
BUFG	5	5	5	5	32
Rate (MBps)	300	1,200	2,400	4,800	–
Rate (MSps)	1,200	4,800	9,600	19,200	–

can be reached, in megabytes per second (MBps) and megasymbols per second (MSps), when the operating frequency is 300 MHz.

5.4. Emulating a NAND Flash channel

As shown in Fig. 6, the NVM-E core can be used to emulate not only single NVM chips, but also more complex configurations. For example, multiple NAND Flash chips, forming a single channel, sharing the same data lines and operating on a pipeline fashion can be emulated using a single NVM-E core. In this case, the address space of the internal DRAM is partitioned between the different NAND Flash targets. Furthermore, the number of data buffers is increased in order to store the page data of all targets simultaneously. Finally, an arbiter is utilized to schedule the write/read accesses to the external DRAM DIMM. The NAND Flash emulator has been implemented on a board containing the above-mentioned Virtex-7 FPGA chip and two 8 GB DDR3 DRAMs. The implemented design consists of two NVM-E cores, where each one emulates a channel consisting of four 2 GB MLC NAND Flash ICs. However, depending on the installed DRAM on the FPGA board, NAND Flash devices with larger storage capacity can be emulated.

Since the value of t_{pg} is approximately an order of magnitude larger than the value of t_{rd} , the number of targets, sharing a NAND Flash channel, can be high and the maximum value is determined by the page size, the data transfer rate (DTR) and t_{pg} . However, other limitations, such as the parasitic capacitances at the signal lines, play an important role on the data integrity and hence, on the maximum number of targets that can share the channel, especially when the rate of the data bus increases. Contrary to a typical NAND Flash channel, where the data bus is shared between all chips, in the NVM-E channel the DQ data bus drives only one NVM-E core, independently to the number of chips being emulated and hence, parasitic capacitances do not affect the operating frequency of the data bus when the number of emulation targets becomes larger. Moreover, although the number of chips being emulated has a minimal impact on the resource usage due to the additional data buffers, it has no effect on the maximum operating frequency of the NVM-E. However, if a small FPGA is used, a large number of emulation targets could reach the limit of the available resources and decrease the operating frequency.

Table 3
Pipeline depths for various NAND Flash technologies.

Interface	Cell	t_{rd} (μ s)	t_{pg} (μ s)	Page size (Bytes)	DTR (MBps)	t_{DT} (μ s)	RPD	WPD	SRTR1 (MBps)	SRTR4 (MBps)	SRTR8 (MBps)	SPTR1 (MBps)	SPTR4 (MBps)	SPTR8 (MBps)
ONFI 1.0	SLC	60	800	2,112	40	52.8	2	16	18.7	40.0	40.0	2.5	9.9	19.8
ONFI 1.0	SLC	60	800	4,224	40	105.6	2	9	25.5	40.0	40.0	4.7	18.7	37.3
ONFI 2.0	MLC	50	900	4,320	166	26.0	3	36	56.8	166.0	166.0	4.7	18.7	37.3
ONFI 2.0	MLC	25	200	4,320	166	26.0	2	9	84.7	166.0	166.0	19.1	76.5	152.9
ONFI 2.0	MLC	50	1,300	8,640	166	52.0	2	26	84.7	166.0	166.0	6.4	25.6	51.1
ONFI 2.0	TLC	90	2,400	9,640	166	58.1	3	42	65.1	166.0	166.0	3.9	15.7	31.4
ONFI 2.2	MLC	35	300	8,640	200	43.2	2	8	110.5	200.0	200.0	25.2	100.7	200.0
ONFI 3.0	MLC	50	1,400	16,384	400	41.0	2	35	180.1	400.0	400.0	11.4	45.5	91.0

Although a typical NAND Flash channel is shared by four chips, our design has been designed to support up to 8 targets/channel. Table 3 illustrates the maximum read pipeline depth (RPD) and write pipeline depth (WPD) that can be supported for different NAND Flash technologies and interfaces. Furthermore, we calculated the sustained read transfer rate (SRTR) and the sustained program (write) transfer rate (SPTR), when the channel is configured without pipelining, with 4 stages of pipelining (typical NAND Flash channel) and with 8 stages of pipelining (NAND Flash emulator channel), respectively.

5.5. Emulating SSDs

The architecture of the NAND Flash emulator allows the implementation of even larger configurations. For instance, depending on the available glue logic, multiple instantiations of the aforementioned NAND Flash channel emulator can be implemented on the same FPGA, in order to emulate the storage area of a SSD with multiple NAND Flash channels. For instance, four FPGAs and the respective DRAMs, configured as described in Section 5.4, can be used, in order to emulate the storage area of a 64 GB SSD with 8 NAND Flash channels. Furthermore, different FPGA boards can be used to emulate hybrid SSD-based systems, providing an environment for developing and evaluating architectures related to their lifetime capacity, such as a redundant array of inexpensive disks (RAID) configurations. Fig. 7 illustrates the architecture to emulate a small scale SSD, where the SSD emulation (SSD-E) units are interconnected via a third generation Peripheral Component Interconnect Express (PCIe) switch.

6. Experimental results

6.1. Experimental set-up

The experimental setup, illustrated in Fig. 8, has been implemented using two FPGA boards. One FPGA board contains a NAND Flash controller and an embedded processor, while the second FPGA board contains the NAND Flash emulation core (NVM-E in this figure as explained earlier). The host machine contains a user-friendly environment, allowing the execution of experiments by employing a set of high-level commands. The communication between the host machine and the embedded processor has been developed using the TCP/IP protocol stack and a custom data transfer protocol [21]. The microprocessor processes the commands sent by the host machine and initiates commands execution at the NVM-E via the NVM controller. The NVM controller was developed as an AMBA AXI4 peripheral and implements data acquisition and logic to interface with the NVM-E core. Both systems have been developed using the Xilinx Vivado 2014.3 suite and implemented on Zynq-7000 zc706 Xilinx evaluation boards, interconnected with special cabling via the HPC FMC connectors.

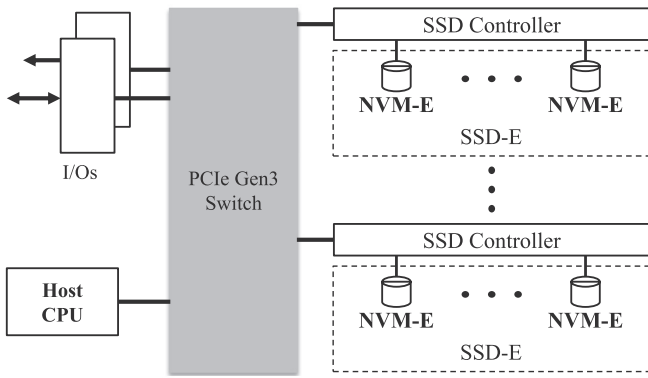


Fig. 7. Emulating hybrid SSD-based systems.

Experimental loading scenarios and analysis have been performed using the MATLAB environment at the host side.

6.2. Emulating commercial MLC and TLC NAND Flash memories

This section presents the experimental results of the NAND Flash core, when emulation of MLC and TLC NAND Flash memories is performed. The internal characteristics of the memories, such as threshold voltage distributions and their variations as a function of their aging state, were unknown. We assumed that the effect of the mean voltage drift was negligible compared to the effect of the standard deviation and therefore during the emulation process we kept the mean noise value equal to zero. This fact was compensated by adjusting the noise's standard deviations as the number of P/E cycles was increased. For the threshold voltage distribution model we used the values $k_1 = 4$ and $k_2 = 2$.

The first step was to determine the BER as a function of the noise's standard deviation. Initially we developed a mathematical model to study this relation [6] and then we performed a set of experiments at the NAND Flash emulator by programming with random data and reading the pages of various blocks, as the normalized standard deviation was increased. The relationship between BER and the normalized standard deviation, when emulating an MLC NAND Flash memory (asymmetric-4PAM), is illustrated in Fig. 9(a).

The next step was to determine the relationship between BER and P/E cycles of the real NAND Flash memory. This process was performed by erasing, programming with random data and reading the pages of various memory blocks, while the raw BER was computed at each P/E cycle. Due to the fact that the experimental data

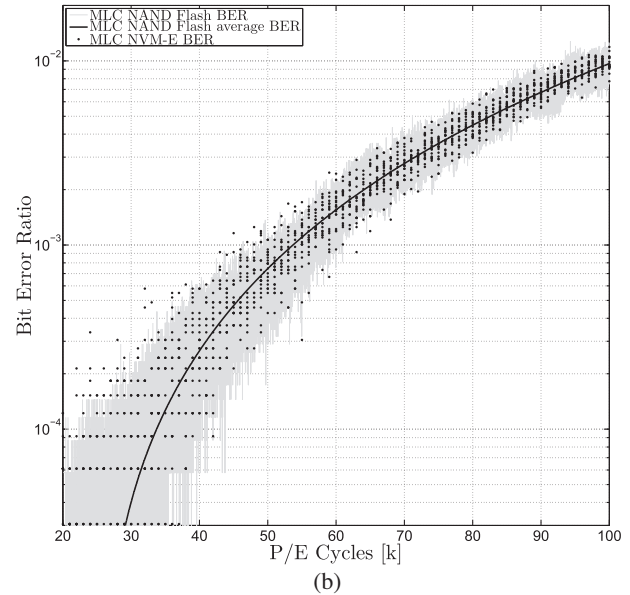
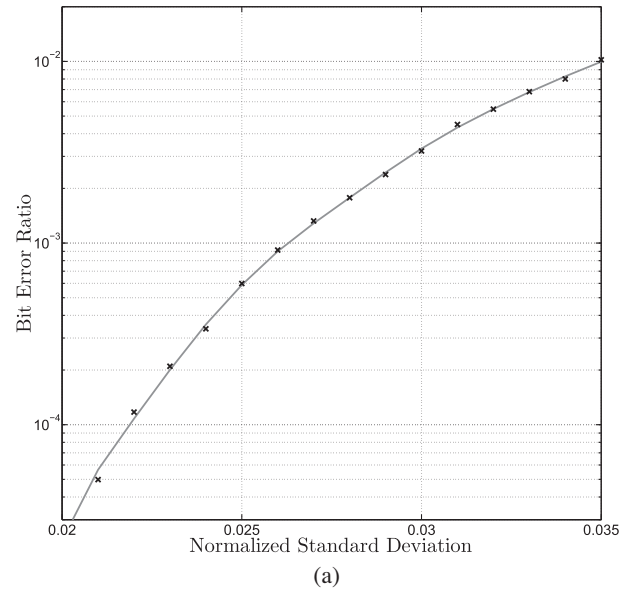


Fig. 9. BER as a function of (a) standard deviation and (b) P/E cycles, using an MLC NAND Flash device and the MLC NAND Flash emulator.

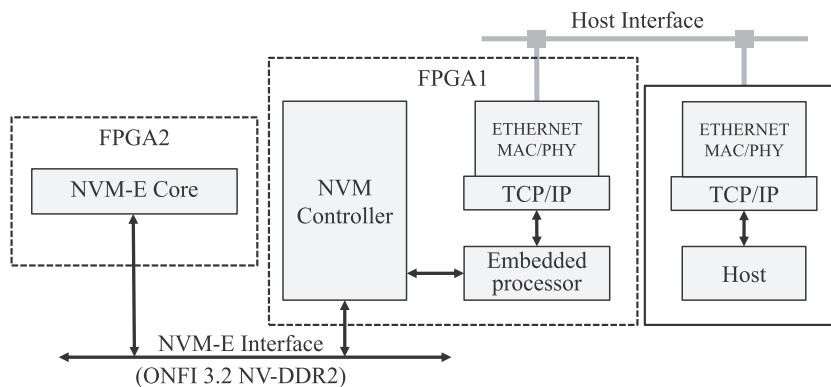


Fig. 8. Experimental setup.

Table 4
Comparison between NVM-E and NAND Flash memory measurements.

P/E cycles (K)	20	40	60	80	100
NVM-E mean ($\times 10^{-3}$)	0.03	0.30	1.63	4.49	9.69
NAND Flash mean ($\times 10^{-3}$)	0.01	0.33	1.46	4.50	9.23
NVM-E variance ($\times 10^{-6}$)	0.00	0.02	0.13	0.44	0.88
NAND Flash variance ($\times 10^{-6}$)	0.00	0.01	0.05	0.36	1.08

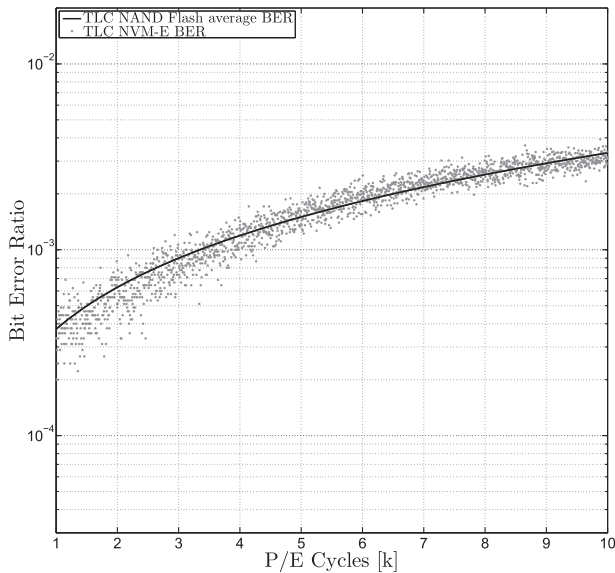


Fig. 10. BER as a function of P/E cycles, using a TLC NAND Flash device and the TLC NAND Flash emulator.

have significant fluctuations, the BER curve was approximated by data fitting. In Fig. 9(b) we present the measurements of BER as a function of P/E cycles of the MLC NAND Flash chip. The experimental BER results are indicated with gray, while the solid black curve represents the average BER.

Using the two curves of Fig. 9(a) and (b), we determined the relation between standard deviation and P/E cycles for the whole lifetime of a device. For a given BER value in Fig. 9(a) we determine the standard deviation and for the same BER value in Fig. 9(b) we find out the respective P/E cycles value. Therefore, the Aging Logic of the NAND Flash emulator was configured with the parameters estimated during the above mentioned process. Then, for various aging conditions we collected data from the NAND Flash emulator using the same procedure as in the NAND Flash device. A set of

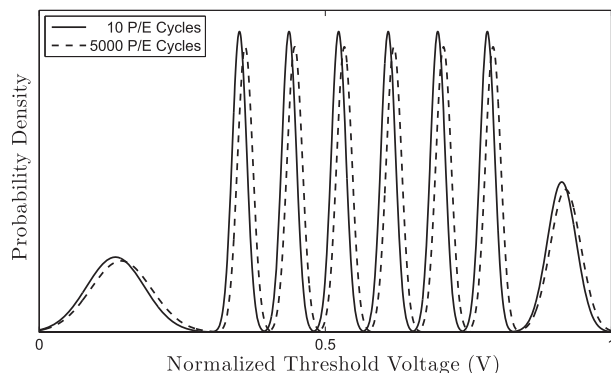


Fig. 11. Threshold voltage distributions of an emulated TLC NAND Flash for two aging conditions.

commands was applied (block erase, programming all pages of the block with random data and reading them back) and BER statistics were collected. The NAND Flash emulator BER measurements are marked with dots in Fig. 9(b).

Table 4 provides a quantitative evaluation of the similarity between the BER results of the NAND Flash emulator and the respective BER results collected from the NAND Flash memory for different aging phases.

A similar approach was followed for the TLC NAND Flash memory. After determining the second order polynomial relationship between σ and the P/E cycles, we configured the Aging Logic of the NAND Flash emulator and then we measured BER. Fig. 10 depicts the average BER measurements stated in [19] along with the BER measurements collected using NVM-E. The distributions of the initial and of an aged state of the emulated TLC NAND Flash memory are shown in Fig. 11. The qualitative voltage threshold shifting for the aged state is based on the qualitative figure for voltage threshold shifting in NAND Flash memories, presented in [22].

Comparing the experimental results of the real NAND Flash devices with the results generated by the presented NAND Flash emulator, it becomes obvious that the proposed architecture represents accurately the behavior of real NAND Flash devices, when configured with the appropriate parameters.

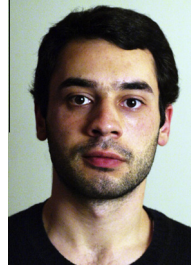
7. Conclusions

The architecture and the functionality of a NAND Flash emulator was analyzed and presented. The NAND Flash emulator can accurately represent the bit error characteristics and the response times of a real NAND Flash chip during its whole lifetime, by associating aging conditions with emulator's internal parameters. The presented emulator can be used for replacing single chip or a set of chips organized as a single memory channel. The emulator's functionality and accuracy was validated by comparing its output in terms of BER statistics with experimental data from real NAND Flash memories. The NAND Flash emulator provides a valuable tool for development and evaluation of memory-related algorithms, since it offers real-time and high precision emulation under user-defined aging conditions and adjustability to the characteristics of the emulated NAND Flash technology.

References

- [1] B. Chen, X. Zhang, Z. Wang, Error correction for multi-level NAND flash memory using Reed-Solomon codes, in: IEEE Workshop on Signal Processing Systems (SiPS), 2008, pp. 94–99.
- [2] F. Sun, K. Rose, 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, 2006.
- [3] Z. Wang, M. Karpovsky, A. Joshi, Reliable MLC NAND flash memories based on nonlinear t-error-correcting codes, in: 2010 IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2010, pp. 41–50.
- [4] W. Xu, T. Zhang, A. Time-Aware, Fault tolerance scheme to improve reliability of multilevel phase-change memory in the presence of significant resistance drift, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 19 (8) (2011) 1357–1367.
- [5] H. Pozidis, N. Papandreou, A. Sebastian, T. Mittelholzer, M. BrightSky, C. Lam, E. Eleftheriou, Reliable MLC data storage and retention in phase-change memory after endurance cycling, in: 2013 5th IEEE International Memory Workshop (IMW), 2013, pp. 100–103.
- [6] A. Prodromakis, G. Sklias, T. Antonakopoulos, Emulating the aging of NAND Flash memories as a time-variant communications channel, in: 2014 6th International Symposium on Communications, Control and Signal Processing (ISCCSP), May 2014, pp. 278–281.
- [7] S. Korkotsides, G. Bikas, E. Eftaxiadis, T. Antonakopoulos, BER analysis of MLC NAND flash memories based on an asymmetric PAM model, in: 2014 6th International Symposium on Communications, Control and Signal Processing (ISCCSP), May 2014, pp. 558–561.
- [8] A. Prodromakis, S. Korkotsides, T. Antonakopoulos, A versatile emulator for the aging effect of non-volatile memories: the case of NAND flash, in: 2014 17th Euromicro Conference on Digital System Design (DSD), August 2014, pp. 9–15.

- [9] J. Brewer, M. Gill, *Nonvolatile Memory Technologies with Emphasis on Flash: A Comprehensive Guide to Understanding and Using Flash Memory Devices*, vol. 8, Wiley.com, 2011.
- [10] R. Bez, E. Camerlenghi, A. Modelli, A. Visconti, Introduction to flash memory, *Proc. IEEE* 91 (4) (2003) 489–502.
- [11] K. Kanda, N. Shibata, T. Hisada, K. Isobe, M. Sato, Y. Shimizu, T. Shimizu, T. Sugimoto, T. Kobayashi, N. Kanagawa, Y. Kajitani, T. Ogawa, K. Iwasa, M. Kojima, T. Suzuki, Y. Suzuki, S. Sakai, T. Fujimura, Y. Utsunomiya, T. Hashimoto, N. Kobayashi, Y. Matsumoto, S. Inoue, Y. Suzuki, Y. Honda, Y. Kato, S. Zaitso, H. Chibvongodze, M. Watanabe, H. Ding, N. Ookuma, R. Yamashita, A 19 nm 112.8 mm² 64 GB multi-level flash memory with 400 Mbit/sec/pin 1.8 V toggle mode interface, *IEEE J. Solid-State Circ.* 48 (1) (January 2013) 159–167.
- [12] Open NAND Flash Interface Specification, ONFI Workgroup, (v3.0). <<http://www.onfi.org/specifications>>.
- [13] K. Prall, Scaling non-volatile memory below 30 nm, in: 22nd IEEE Non-Volatile Semiconductor Memory Workshop, August 2007, pp. 5–10.
- [14] Yu Cai, Erich F. Haratsch, Onur Mutlu, Ken Mai, Threshold voltage distribution in MLC NAND flash memory: characterization, analysis, and modeling, in: Design, Automation & Test in Europe Conference Exhibition (DATE), 2013, pp. 1285–1290.
- [15] D. hwan Lee, W. Sung, Estimation of NAND flash memory threshold voltage distribution for optimum soft-decision error correction, *IEEE Trans. Signal Process.* 61 (2) (2013) 440–449.
- [16] G. Atwood, A. Fazio, D. Mills, B. Reaves, Intel StrataFlash™ memory technology overview, *Intel Technol. J.* (1997).
- [17] S. Li, T. Zhang, Improving multi-level NAND flash memory storage reliability using concatenated BCH-TCM coding, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* 18 (10) (2010) 1412–1420.
- [18] Y. Maeda, H. Kaneko, Error control coding for multilevel cell flash memories using nonbinary low-density parity-check codes, in: 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT '09), 2009, pp. 367–375.
- [19] E. Yaakobi, L. Grupp, P. Siegel, S. Swanson, J. Wolf, Characterization and error-correcting codes for TLC flash memories, in: 2012 International Conference on Computing, Networking and Communications (ICNC), January 2012, pp. 486–491.
- [20] G.E.P. Box, M.E. Muller, A note on the generation of random normal deviates, *Ann. Math. Statist.* 29 (2) (1958) 610–611.
- [21] N. Papandreou, T. Antonakopoulos, U. Egger, A. Palli, H. Pozidis, E. Eleftheriou, A versatile platform for characterization of solid-state memory channels, in: 2013 18th International Conference on Digital Signal Processing (DSP), 2013, pp. 1–5.
- [22] L. Crippa, R. Micheloni, I. Motta, M. Sangalli, Nonvolatile memories: NOR vs. NAND architectures, in: R. Micheloni, G. Campardo, P. Olivo (Eds.), *Memories in Wireless Systems, Signals and Communication Technology*, Springer, Berlin Heidelberg, 2008, p. 51.



research interests are non-volatile memories and error correction techniques.



currently a Professor and the Head of the Laboratory of Electrotechnics. From 2001 to 2002, he spent his sabbatical at the IBM Zurich Research Laboratory. His research interests are in the areas of communications and embedded systems with emphasis on performance analysis, efficient hardware implementation, and rapid prototyping. He has more than 150 publications in these areas and is actively participating in several R&D projects of European industries. Dr. Antonakopoulos is a senior member of the IEEE and a member of the Technical Chamber of Greece.



Antonios Prodrumakis received the diploma (B.S. and M.S. degrees) in electrical and computer engineering and the M.S. degree in integrated software and hardware systems from the University of Patras, Patras, Greece in 2012 and 2014, respectively. He is currently a Ph.D. student, researcher and developer of integrated software and hardware systems in the Communication and Embedded Systems group, Department of Electrical and Computer Engineering, University of Patras. His research interests include the development and characterization of solid-state storage devices and systems, as well as neuromorphic systems and encoding techniques.