The Effect of Using Multiple Code Rates on NVM-based Storage Systems

Stelios Korkotsides and Theodore A. Antonakopoulos
Department of Electrical and Computer Engineering
University of Patras
Patras 26504, Greece
Email: stelkork@ece.upatras.gr, antonako@upatras.gr

***Abstract*—Non-Volatile Memories (NVM) continue to increase their density in order to achieve higher storage capacity, but require more powerful Error Correction Codes (ECC) to cover the need for data reliability. Low Density Parity Check (LDPC) codes provide a viable solution to this problem, at the cost of higher complexity and power consumption, which constraints most NVM devices to use a fixed LDPC rate. In the following paper, we present an architecture of a NVM based storage system which maximizes its lifetime by dynamically adapting the LDPC's rate according to the aging condition of its devices. Furthermore, we have developed a queuing model in order to study the effects of different implementation choices on the system's IO performance.**

I. Introduction

Non-Volatile Memories (NVM) provide a vital solution to the storage requirements of consumer devices and have already been used in data centers and other enterprise systems. The main technologies used in Solid State Drives (SSDs) are NAND Flash and PCM. They both have the advantages of high I/O performance, low power consumption and small size, but in order to provide enough capacity to cover the contemporary market's needs, aggressive increase of storage density via technology scaling has to be performed. It is well known that this density increase has inevitably come with some side-effects, like declining reliability, endurance and write performance [1]. Consumer applications are quite tolerable to these problems, but enterprise systems cannot sacrifice their reliability at any cost.

This reliability is achieved mainly by using Error Correction Codes (ECC). SSD controllers include a module that encodes and decodes user data usually using block ECCs, either Bose Chaudhuri Hocquenghem (BCH) or/and Low Density Parity Check (LDPC) [3]–[5]. BCH codes have been extensively used in the past, due to their low implementation complexity, but the superiority of the LDPC codes in error correction capability is turning them into the mainstream ECC solution for future products [6]–[8]. Currently, LDPC codes can cover sufficiently the rising need for stronger ECC, due to the above mentioned scaling side-effect.

LDPC error correction capability is adequate for most storage application, but the use of LDPC codes increases the system complexity and affects the system's I/O performance for a given target reliability. In early stages of SSD's life, data have very low raw BER, but after several hundreds or thousands of program/erase (PE) cycles, raw BER increases and the decoder needs more iterations to successfully correct the errors. In addition, for a given user reliability (user BER) the lifetime of the device can be further increased by using stronger ECC at the expense of higher overhead, due to lower code rate. This reduces the user capacity and thus partially offsets the advantages of technology scaling. Most SSDs use a fixed code rate which varies between 0.75 and 0.95, but they sacrifice user space at the early life of the SSD or their error correction ability collapses earlier than using a stronger code. A more complex model of adaptive code rate LDPC decoder satisfies adequately the trade-off between SSD capacity and ECC performance, at the expense of higher implementation complexity.

In this paper we use the architecture of a NVM-based storage system [9] and we present a queuing model in order to investigate the performance of the proposed architecture on different workloads and structural variations. The proposed architecture is able to almost double the system's lifetime capacity compared to fixed LDPC rate approaches, while at the same time it provides guaranteed reliability, keeps the total implementation complexity relatively low and achieves high I/O performance.

The idea of using multiple code rates for both BCH and LDPC is described in [10]–[13]. In [14] proposed an ECC scheme with adaptive strength of BCH codes, by lengthening the codewords instead of switching rates and thus the user capacity remains the same during the lifetime of the device. In our approach the decoding units are not internal components in each SSD but form a pool of available decoders. A main storage system controller arbiters the user requests and manages the pathway of the data to this pool of decoders. Decoders are dynamically assigned to data originating from any SSD in the system. The use of external LDPC decoders helps us to lower the cost of the SSDs, implement more complex coding schemes, such as adaptive rate, and therefore increases the lifetime capacity of the devices, while decreasing power consumption and the whole system complexity.

Section II presents the proposed architecture and describes the system's model. In Section III we present the effect of using multiple code rates on the system's lifetime and we demonstrate the advantages of the proposed approach. Finally, in Section IV we present the system's I/O performance for different structural and implementation parameters.

II. NVM-BASED STORAGE SYSTEM

A common enterprise storage system consists of a number of SSDs, which are attached to the Main Storage Controller (MSC) via high speed links. Each SSD contains a controller that interfaces to the MSC and to a number of NVM channels. Depending on the internal architecture, the SSD controller performs functions like logical-to-physical addressing, wear-leveling, garbage collection etc and contains a control module per NVM channel for supporting various interfaces like ONFI [15]. Error correction operations ensure reliability in data recovery and are performed either at the SSD level or at the channel level. Due to complexity issues, the second choice is preferable in case of low complexity codes, such as BCH or fixed rate LDPC. When more complicated decoding schemes are used, a feasible option in terms of power consumption and complexity is the use of an ECC block per SSD, available to all the NVM channels. That results to lower I/O rates and it is a compromise between performance and implementation complexity. For the rest of this paper we assume that a concatenated coding scheme is used. An internal BCH code is used in order to correct a small number of errors, while the main decoding is performed by an outer LDPC code. Since the LDPC's performance determines the lifetime of the device and the I/O performance, we are going to deal only with this family of codes.

Fixed rate ECCs do not take into account the non-linear relationship between aging conditions and the target user BER requirements. At the beginning of the lifetime of the device a weak code would be sufficient for the number of introduced errors. As time progresses and the device's aging affects its reliability, a stronger code has to be employed in order to increase its error correction capability according to the target user BER. Consequently, a weak code would offer large user storage space for less lifetime of the device, while a stronger code would increase the lifetime but sacrifice more storage space. A variable ECC scheme though, would use the advantages of both cases and offer an optimum solution, at the expense of higher complexity.

Due to the complexity introduced, the variable ECC scheme would not be viable if an ECC block was used in each NVM channel. The use of one ECC block per SSD would be a more reasonable solution, but we follow another approach which offers better flexibility, as it will be shown in the experimental results. The architecture of the proposed storage system is shown in Fig. 1. The ECC blocks were dismounted from the SSDs and are used at a system basis, that is, a number of N decoders are connected to the PCIe switch and they are dynamically shared by all SSDs. When a read command is applied by the host via the High Speed I/O Link, the Main Storage System Controller passes it to the corresponding SSD. The data are retrieved by the NV memories and if errors have been detected, an idle LDPC decoder is selected for error correction. LDPC decoders are dedicated hardware accelerators, such as FPGA boards or GPUs, that can perform simultaneous decoding of multiple LDPC codewords using multiple code rates in order to increase their throughput. In such a storage system, SSD controllers track the aging condition of their NVM chips and they adapt the ECC dynamically throughout their lifetime.

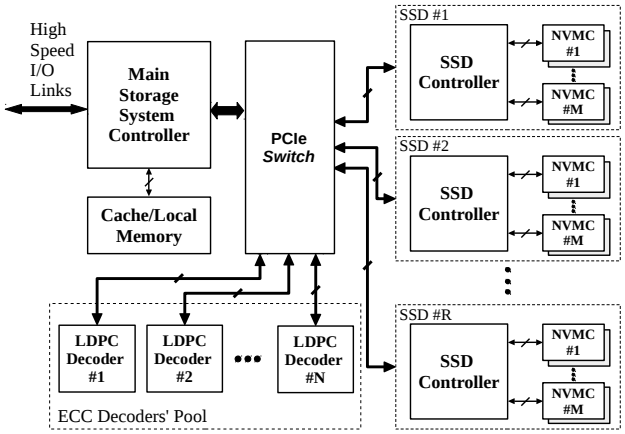

Fig. 1. Storage system architecture.

In order to investigate the effect of the numbers of LDPC decoders and the decoders' delay on the system's I/O performance, we developed the queuing model that is shown in Fig. 2. User write requests are distributed over R SSDs, using either a round-robin scheme or a custom allocation scheme. Respectively the requests in each SSD are guided to one of the M NVM channels (NVMC), delayed for an amount of time that represents the chip access time. Consequently read requests follow the same traffic pattern. Depending on the age condition of the devices, each request is labeled with a number of iterations needed to recover the initial data, using probability distributions that have been obtained by executing the LDPC algorithm in several simulated aging states of the devices. The requests are assigned to one of the decoders, unless they are error free, in which case they are sent directly to the output. In each of the above steps, the requests are stored in queues (FIFOs), until the respective decoder is available. When all buffers in the path of interest are full, the user is blocked from sending more requests, until there is space available in the system.

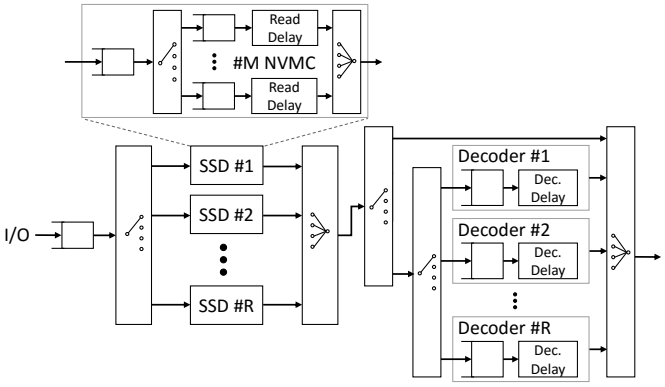

Fig. 2. Queuing model of the proposed storage system.

III. The use of multiple LDPC codes

As mentioned earlier, the use of multiple ECC rates is necessary in order to increase the system's lifetime capacity. Each code rate is related with the storage system's aging conditions, so, in that sense, it is an adaptive rate system. We use the parameters shown in Table I as an example in order to quantify the advantages of this adaptive rate system. The total raw capacity is 8TB, but the user can only access part of it, depending on the data rate used and the data partitioning scheme used. User data are split into blocks of 4KB usually called User Pages (UP). The memory controller splits UPs into an number of Data Blocks (DB, 512 bytes each) in order to form the LDPC datawords (DW). Each DW is composed by a number of DBs. The LDPC codewords (CW) are of fixed size, 8KB in this case. Although CWs are of fixed size, the size of DWs and the number of DBs that each DW contains, depends on the rate of the code used.

Lifetime Capacity (LTC) is a measure of the number of user data that can be written in the storage device throughout its whole life. LTC $=$ (Endurance $\times$ User Capacity)/WAF, where Endurance is the number of P/E cycles that can be performed on the device before the User BER (UBER) exceeds a predefined threshold, User Capacity is the number of bytes that are available to the user and WAF is the Write-Amplification-Factor, which is associated with internal SSD techniques like wear-leveling, garbage collection start-gap, etc.

The performance of the system depends greatly on the number of decoding iterations. The adaptive system dynamically switches between the rates $5/6$, $3/4$, $2/3$, $1/2$, $1/3$ and $1/4$, targeting a user BER (UBER) better than 10^{-14} and a maximum of 10 iterations as shown in Fig. 3. Each rate r_i has a limit of PE_i cycles for $UBER \geq 10^{-14}$. Fig. 3 shows the mean number of iterations. In reality though, this number varies depending on the exact number of data errors and their position inside the codewords. Consequently, the number of iterations is determined by probability distributions as shown in Fig. 4 for LDPC rate $1/4$. The distributions are similar for all used LDPC codes at different aging states.

Detailed description and results of the gains in lifetime capacity can be found in [9]. In order to provide the reader with a taste of the benefits of the adaptive LDPC rate system, we present in Table II only the resulting increase in lifetime for sustained data rates compared to fixed LDPC rate systems.

As shown in Fig. 5 the performance of each code diminishes as the device ages, due to the increasing number of iterations needed to decode successfully a codeword. By switching into stronger codes when LDPC decoder's iterations pass a predetermined limit, the performance decrease can be decelerated and kept relatively high throughout the lifetime of the device until all LDPC codes have been used.

IV. Queuing Model and Simulation Results

The model of Fig. 2 was developed in Matlab's SimEvents® discrete event simulation software in order to estimate the system performance. The data access time of NVM chips is considered to be $t_{dat} = 80$us, with 40us for internal operations and 40us of data transfers. In case of multiple chips per channel, data are multiplexed in the data bus and the access time of the channels as observed by the host is 40us. The rest of the SSD parameters are shown in Table I. In the following measurements we varied the location, the number of LDPC decoders used and their decoding time in order to determine how the number of decoders and the decoding time per iteration affect the system performance throughout its lifetime. The memory aging determines the number of iterations needed to recover the data and consequently it affects the data rate and the system's latency. The number of iterations is estimated from probability distributions similar to Fig. 4 and the system switches between the LDPC rates $r_i = \{5/6, 3/4, 2/3, 1/2, 1/3, 1/4\}$.

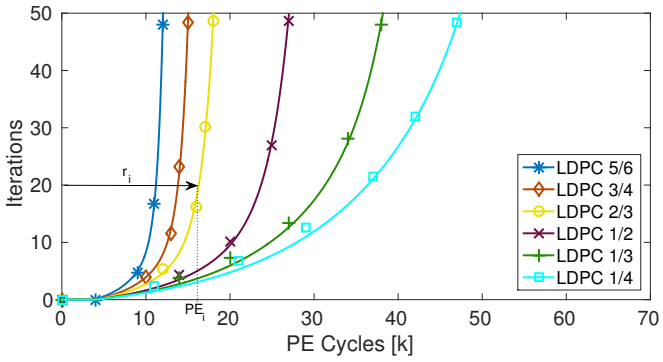

Fig. 3. Number of decoding iterations per LDPC rate versus PE cycles [9].

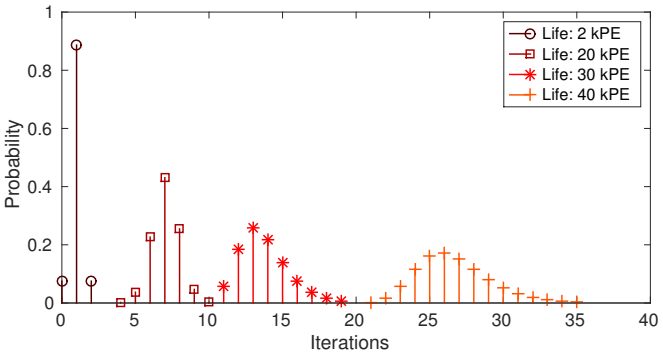

Fig. 4. Decoding iterations for various NVM aging conditions.

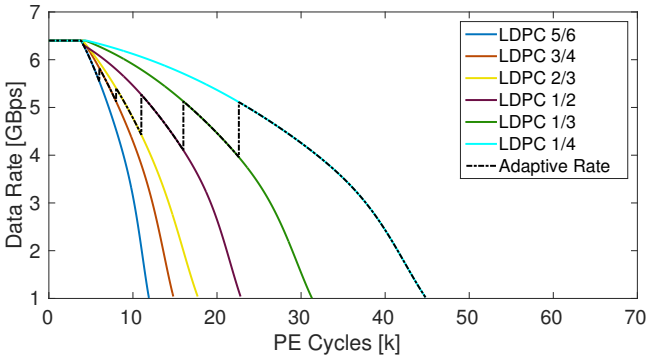

Fig. 5. Evolution of I/O Rate versus PE cycles for various LDPC codes [9].

TABLE I
NON VOLATILE SYSTEM PARAMETERS

NVM Chip Specs		Storage System Specs	
Capacity [Gbits]	512	Chips per Channel	4
Page [Bytes]	16384	Channels per SSD	8
Pages per Block	256	Number of SSDs	8
Number of Blocks	16384	**Total Capacity [TB]**	**32**

TABLE II
SYSTEM LIFETIME FOR SUSTAINED DATA RATES

Code Rate	Lifetime [Years]	Endurance [kPE Cycles]	Normalized Data Rate [GBps]
5/6	0.92	10	2.6
3/4	1.15	14	3.1
2/3	1.16	17	3.6
1/2	1.50	25	3.6
1/3	1.31	34	5.3
1/4	1.20	43	8.0
Adaptive	2.30	43	8.4

In all simulation results, we use the factor $F = \frac{MR}{N}$, where M is the number of NVM channels per SSD, R is the total number of SSDs and N is the number of LDPC decoders.

A. Scenario 1

In the first simulation scenario, we use $MR = 64$ and we vary the number of decoders, in order to study its effect on system's performance. The internal decoding time per iteration was set to 4 usecs. In real systems, this time varies slightly between different code rates, but the variation is insignificantly small and does not affect the performance results.

Since every request always passes through one NVM chip and one decoder, the number and the location (shared in a pool or internal to the SSDS) of the LDPC decoders of the system does not affect the latency as shown in Fig. 6. As expected the latency increases as the system becomes older, since a larger number of iterations is required for fully decoding a codeword. On the other hand, the maximum data rate is severely affected by N as shown in Fig. 7, but their position inside the system does not make significant difference. We measure the data rate in KIOPs, where an IOP refers to a 8kB codeword.

At this point we have to note that the above measurements were taken by distributing the user requests uniformly over all SSDs, which is the case that provides the maximum data rate, since all SSD devices and LDPC decoders are fully utilized. When there is one LDPC decoder per SSD or one in each NVM channel, the data rate is equal to the data rate of a system that uses a pool of 8 or 64 decoders respectively.

In Fig. 8 we show the performance difference between using one dedicated decoder per SSD and a pool of 8 globally shared decoders when accessing data not equally distributed. In this scenario we use two types of data, hot (more frequently accessed) and cold data, and the hot data are located in two

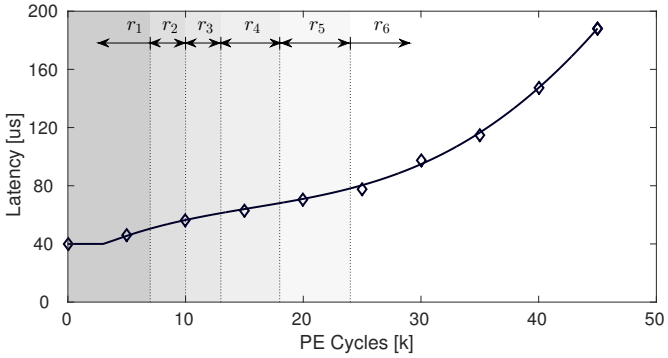

Fig. 6. Scenario 1: System latency for uniformly data accessing

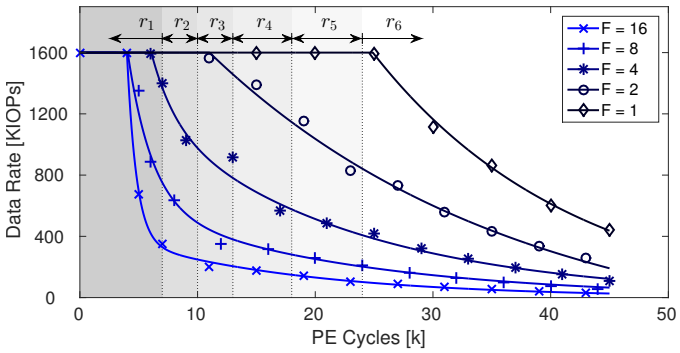

Fig. 7. Scenario 1: System data rate for uniformly data accessing

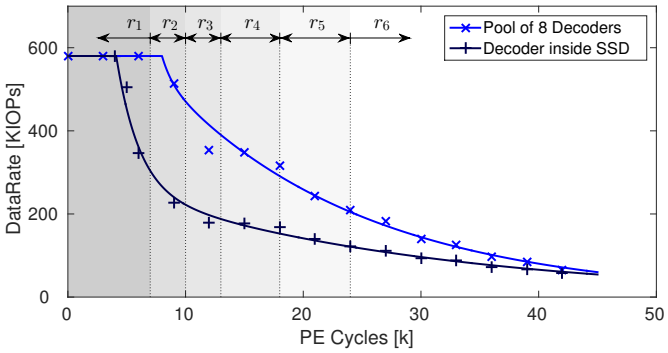

Fig. 8. Scenario 1: System data rate for cold/hot data accessing

SSDs. In this case we notice a major improvement in data rate when we use the pool of decoders.

B. Scenario 2

Another important factor for the system is the performance of the decoder, expressed via decoding time per iteration t_{iter}. This factor is affected by the technology of hardware accelerators used and the efficiency of the decoding algorithm implementations. For this reason, we define $T = \frac{t_{dat}}{t_{iter}}$. Since the measure of study is t_{iter}, we keep the data access time fixed at $t_{dat} = 40us$ as in the previous scenario. For the following measurements we used a pool of 8 shared decoders. In Fig. 9 is shown the latency of the system for various values

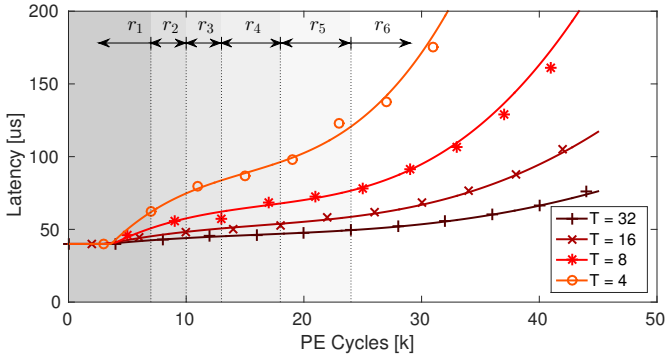

Fig. 9. Scenario 2: System latency for various iteration delays

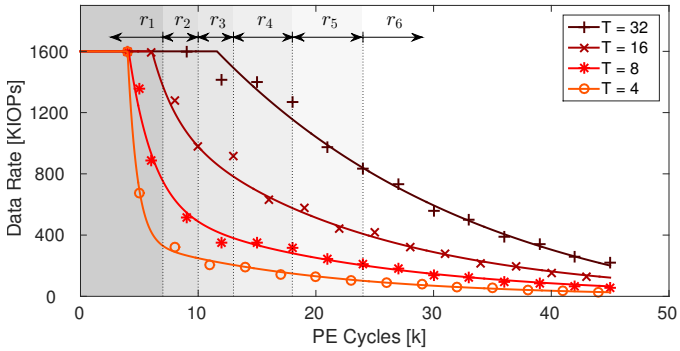

Fig. 10. Scenario 2: System data rates for various iteration delays

of T, while in Fig. 10 we depict the system's data rate in 8kB IOPs.

V. Conclusions

In this work we studied the performance of an enterprise NVM-based storage system that uses multiple LDPC code rates adapted to the aging conditions of its SSDs. The simulation results indicate that the use of the adaptive rate almost doubles the lifetime of the devices and the position of the LDPC decoders in a shared pool enhances the system's performance comparing to the dedicated use per SSD or NVM channel when the data are not uniformly distributed to all SSDs. The above architecture is targeting enterprise applications and its performance can be fine-grained customized according to the specific user needs.

References

[1] Laura M. Grupp, John D. Davis and Steven Swanson, *"The Bleak Future of NAND Flash Memory"*, in Proceedings of the 10th USENIX Conference on File and Storage Technologies, 2012, pp. 2-2

[2] Yu Cai, Erich F. Haratsch, Onur Mutlu and 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, 18-22 March 2013

[3] Youngjoo Lee, Hoyoung Yoo, Injae Yoo, In-Cheol Park, *"6.4Gb/s multi-threaded BCH encoder and decoder for multi-channel SSD controllers"*, Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International, pp.426-428, 19-23 Feb. 2012.

[4] T. Tokutomi, S. Tanakamaru, T.O. Iwasaki, K. Takeuchi, *"Advanced error prediction LDPC for high-speed reliable TLC nand-based SSDs"*, IEEE 6th International Memory Workshop (IMW) 2014, pp.1-4, 18-21 May 2014.

[5] Binbin Li, Bolun Zhang, Yifan Zhang, Dongmei Xue, *"Use soft-decision error-correction codes in Phase-Change Memory"*, in Semiconductor Technology International Conference (CSTIC), 2015 China, pp.1-3, 15-16 March 2015.

[6] R.C. Bose and D.K. Ray-Chaudhuri, *"On A Class of Error Correcting Binary Group Codes"*, Information and Control, 1960, vol. 3, pp. 68–79.

[7] Robert G. Gallager, *"Low-Density Parity-Check Codes"*, 1963.

[8] Kai Zhao, Wenzhe Zhao, Hongbin Sun, Xiaodong Zhang, Nanning Zheng and Tong Zhang, *"LDPC-in-SSD: Making Advanced Error Correction Codes Work Effectively in Solid State Drives"*, in 11th USENIX Conference on File and Storage Technologies (FAST 13), 2013, pp. 243–256.

[9] Stelios Korkotsides and Theodore A. Antonakopoulos, *"Architecture of a NVM-based Storage System Using Adaptive LDPC Codes*, Modern Circuits and Systems Technologies (MOCAST), 2016, 12 - 14 May 2016.

[10] S.Hellmold, *"The evolving NAND flash business model for SSD"*, in Proceedings of flash memory summit, Santa Clara, 2010.

[11] Jen-Wei Hsieh, Chung-Wei Chen and Han-Yi Lin, *"Adaptive ECC Scheme for Hybrid SSDs"*, IEEE Transactions on in Computers, vol.64, no.12, pp.3348–3361, December 2015.

[12] Stephen Bates, *"Using Rate-Adaptive LDPC Codes to Maximize the Capacity of SSDs"*, in Proceedings of Flash Memory Summit, Santa Clara, 2013.

[13] Shigui Qi, Dan Feng, Nan Su, Wenguo Liu and Jingning Liu, *"A New Solution Based on Multi-Rate LDPC for Flash Memory to Reduce ECC Redundancy"*, in IEEE Trustcom/BigDataSE/ISPA, 2015, vol.1, pp.918–923, 20-22 August 2015.

[14] Shuhei Tanakamaru, Mayumi Fukuda, Kazuhide Higuchi, Atsushi Esumi, Mitsuyoshi Ito, Kai Li, Ken Takeuchi, *"Post-manufacturing, 17-times acceptable raw bit error rate enhancement, dynamic codeword transition ECC scheme for highly reliable solid-state drives, SSDs"*, Solid-State Electronics, Volume 58, Issue 1, April 2011, pp. 2–10.

[15] Y. J. Seong, E. H. Nam, J. H. Yoon, H. Kim, J. y. Choi, S. Lee, Y. H. Bae, J. Lee, Y. Cho and S. L. Min, *"Hydra: A Block-Mapped Parallel Flash Memory Solid-State Disk Architecture"*, in IEEE Transactions on Computers, vol. 59, no. 7, pp. 905–921, July 2010.