Architecture of a NVM-based Storage System Using Adaptive LDPC Codes

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*—Low Density Parity Check (LDPC) codes have been widely used in communications systems due to their high error correction capabilities. Recently these codes are also investigated for being exploited in high performance storage systems, especially when Non-Volatile Memory (NVM) technologies are used. The main drawback of using LDPC codes in storage systems with a high number of parallel channels is the increased hardware complexity and cost, especially when variable rate codes are used. In this work, we present an architecture of a NVM-based storage system that dynamically adapts the LDPC's rate to the aging conditions of the storage device in order to maximize its lifetime capacity while keeping low its hardware complexity. In order to decrease the system's complexity we propose a PCIe-based architecture that uses a pool of LDPC decoders shared by all NVM channels and we study its effect on the system's lifetime capacity and the achievable I/O data rate.**

I. Introduction

The advances on Flash technology in the last decade has turned it into the main technology for storage devices and systems, like Solid-State Drives (SSDs). Although Hard-Disk Drives (HDDs) have lower cost per GB, SSDs continuously increase their market share by providing higher I/O rates and increased capacity. In addition, other non-volatile technologies, such as Phase Change Memory (PCM), have emerged and although they are primarily in experimental stages they provide promising results for being used in commercial and enterprise products in the future. The continuous scaling of these technologies has increased their storage density and now such devices are approaching the capacity of the traditional HDDs. It is well known that NV memories provide low consumption, high data rates and high retention time. However, the scaling of solid state memory technologies and the use of multi-level cells has generated a number of new and challenging technical problems, mainly related with aging effects (expressed as Program-Erase (PE) cycles) and the evolution of raw data errors. Consequently they are related with the maximum system life-time for a given ECC [1].

The most efficient way to improve the reliability and the lifetime of SSDs is to use block Error Correction Codes (ECC), such as BCH and LDPC [2]–[4]. Bose Chaudhuri Hocquenghem (BCH) codes are the most used ECCs in SSDs, due to their low implementation complexity, but Low Density Parity Check (LDPC) codes [5] can outperform them and have started to replace them due to the constantly increasing demand for more powerful codes. Therefore LDPC codes are considered as the main candidate for the future ECC in SSDs. For a given user reliability (user BER), the lifetime of a device can be increased by employing stronger ECC codes at the expense of higher overhead (smaller code rate). This decreases the user capacity of the whole storage device and consequently it will partially offset the advantages of technology scaling. The selection of the proper code rate for a device is a trade-off between SSD's capacity and ECC performance. Most SSDs use code rates between 0.75 and 0.95 which remain fixed for the lifetime of the device.

In this paper we present an architecture of a NVM-based storage system that dynamically switches between multiple LDPC code rates in order to extend its lifetime capacity (LTC). The lifetime of the system can be extended up to four times compared to a fixed-code rate implementation and is mainly targeted enterprise applications that require high reliability during the lifetime of their storage devices. In addition, the power consumption and the complexity of the SSD is decreased by dismounting the LDPC decoder from the SSD controller and creating a pool of decoders. The decoders are dynamically used by various NVM channel (NVMC) controllers when errors have been detected in the recovered data. The use of adaptive code rates is mentioned in [6] as a solution for satisfying the the increased reliability requirements of new applications. [7] presents a BCH ECC architecture for SSDs, where four rates of BCH codes are used and parity bits are stored in dedicated memory chips. An ECC scheme with adaptive strength based also on BCH codes is proposed in [8]. The main advantage of this scheme is that the strength of the code is increased by lengthening the codewords instead of switching rates and thus the user capacity is not decreased during the lifetime of the device. An adaptive rate QC-LDPC code scheme is used to increase the lifetime capacity of SSDs is presented in [9], but the LDPC decoder is a part of the SSD controller, and that increases the system's complexity linearly with the number of memory channels used. Last but not least, in [10] are presented performance and energy consumption results from the use of multi-rate LDPC in Flash memories.

Section II presents the architecture. In Section III we present the effect of variable code rates on the system's lifetime capacity and we demonstrate the advantage of the proposed

approach by comparing the use of fixed and multiple code rates. In addition, we present the effect of the proposed approach on system's I/O performance.

II. NVM-based Storage System

An enterprise storage system contains a number of SSD disks, which are connected to the Main Storage Controller (MSC) via a high speed interconnect technology, like PCIe. Each SSD consists of a controller that communicates with the MSC and a number of NVM channels. Depending on the used NVM technology the SSD's controller performs functions like logical-to-physical addressing, wear-leveling, garbage collection and contains a control module per NVM channel for supporting various interfaces like ONFI. Proper ECC encoders and decoders are used for reliable data recovery during the whole lifetime of the device. LDPC codes are commonly used as the outer code of a two concatenated codes scheme, while BCH is used as the inner code. The BCH is a light code that can correct up to a small number of data errors, while LDPC is the main code and its capability determines the maximum system lifetime. It is considered that LDPCs will soon be the only option for MLC and TLC NVM devices. For the rest of this paper we will only deal with this family of block ECCs, considering also the effect of the inner code in the relation between the aging conditions and the performance of the used LDPC codes.

The most efficient way to correct errors in terms of reliability, user capacity and low latency is to install an ECC block (encoder/decoder) in each SSD's channel. Such an implementation will only be viable in terms of energy consumption and hardware complexity when fixed-rate and low implementation complexity code is used. The drawback of using a fixed rate ECC is that it does not take into account the non-linear relation between the aging conditions and the target user BER specification. At the beginning of the lifetime of a storage device no strong ECC is required, while as the time progresses higher error correction capabilities have to be employed. Using a strong fixed rate ECC results to extended lifetime, but its overhead decreases the storage capacity. If a weaker ECC is used, the storage capacity increases since less overhead is need for the parity symbols, but the system's lifetime becomes shorter for any given target user BER. Therefore the need of using an adaptive to the aging conditions error correction scheme becomes necessary.

The large codeword size and the high complexity for performing its arithmetic operations makes the LDPC decoders a demanding system component in terms of hardware resources, even for fixed rate codes. When multiple code rates have to be supported by the same LDPC decoder, the hardware complexity increases further. Up to now, SSDs usually use a dedicated ECC decoder per channel for achieving minimum latency. This results to reasonable complexity when codes like BCH are used, but this is not the case when LDPC codes are used. Therefore the inclusion of a dedicated LDPC decoder in each NVM channel has to be avoided and another approach has to be followed.

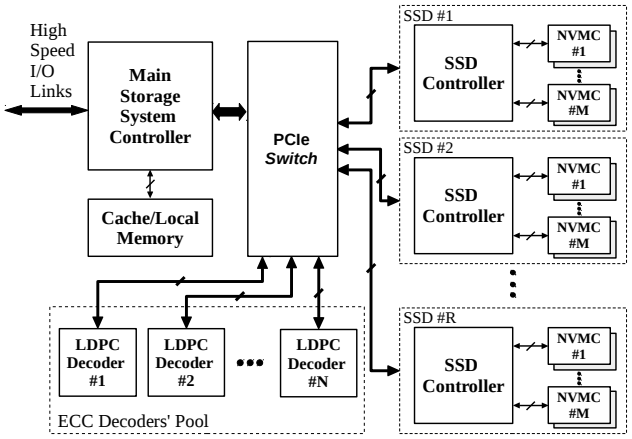

Fig. 1. Block diagram of the proposed storage system with adaptive code rate decoding.

In Fig. 1 we show an alternative architecture for an NVM-based storage system. It includes R SSDs with M NVM channels each. Instead of using MR LDPC decoders we propose to use a smaller number of LDPC decoders in order to compromise between performance and complexity/cost. We assume that there is a pool of N LDPC decoders which can be shared between all SSD controllers. Actually, we considered two approaches. In the first approach each SSD controller contains a dedicated LDPC decoder that can be utilized by its M NVM channel controllers, while the rest $N-M$ LDPC decoders are shared dynamically between all SSDs. In the second approach, no dedicated decoder is used per SSD and all LDPC decoders are shared between all SSDs. This shared pool of LDPC decoders has been interconnected with the SSDs via a PCIe interconnect switch. When a read command is applied by the host via a 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 eg. FPGA boards or GPUs, that can perform simultaneous decoding of multiple LDPC codewords with multiple code rates. In such a storage system, SSD controllers track the aging condition of their NVM chips and they adapt the ECC dynamically throughout their lifetime. The number of LDPC decoders needed for achieving good I/O performance is under investigation and depends on the NVM technology, system configuration (number of SSDs and number of channels per SSD) and the target I/O rate. In any case, it holds that $N \ll MR$.

III. The Use of Multiple LDPC Codes

In order to study the performance of a storage system described previously we use as an example the storage system of Table I. The total raw capacity is 32TB, but depending on the LDPC rate and the data partitioning scheme used, the user experiences a different capacity. Figure 2 shows the allocation of user pages to codewords and NVM pages. The

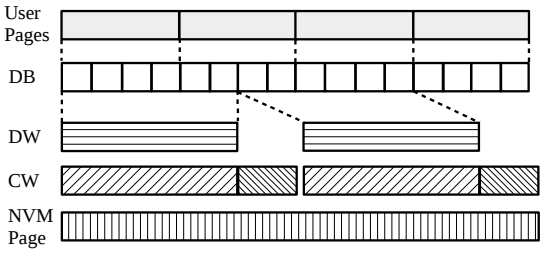

Fig. 2. User data partitioning scheme for code rate $3/4$

user data are split into 4KB blocks 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. 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, as shown in Table II. In our analysis we adopt two methods to map UPs into NVM pages. The first method (M1) splits and allocates a UP into multiple CWs, without mixing CWs from different UPs into the same NVM page. The second method (M2), removes the above restriction and makes better storage space utilization, but makes the data access more complex and increases the average number of I/O accesses, as also shown in Table II.

A. Lifetime Capacity Analysis

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 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. Since WAF is independent of the LDPC codes, for comparative results it is valid to assume WAF $= 1$.

In order to calculate the results of an adaptive code rate scheme, we used six LDPC rates: $5/6$, $3/4$, $2/3$, $1/2$, $1/3$ and $1/4$. Table II shows the System Capacity of each rate for both data partitioning methods. In addition, we present the lifetime capacity of the storage system when fixed LDPC rates are used. The Endurance for each code rate is acquired by Fig. 3, by setting the target UBER to 10^{-14}. Fig. 3 has been generated using experimental results of a state-of-the-art NVM chip [11] and the above mentioned LDPC codes with 64k bits codeword size.

Next we estimate the lifetime capacity of the system when multiple code rates are used. In addition to the target UBER of 10^{-14}, we have set a limit to the number of LDPC iterations before switching to the next LDPC rate. This is shown in Fig. 4. Each rate r_i has a limit of PE_i cycles for $UBER \geq 10^{-14}$. Table III shows the results of 4 iteration thresholds and the improvement factor of the adaptive code rate in terms of lifetime capacity compared to the fixed rate implementation.

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	16
Pages per Block	256	Number of SSDs	8
Number of Blocks	16384	**Total Capacity [TB]**	**32**

TABLE II
SYSTEM AND LIFETIME CAPACITY FOR VARIOUS CODE RATES

Code Rate	**DB/CW**	**UP/(NVM page)**		**User Capacity [TB]**		**LTC [PB]**	
		M1	**M2**	**M1**	**M2**	**M1**	**M2**
5/6	13	3.00	3.25	24	26	257	279
3/4	12	3.00	3.00	24	24	330	330
2/3	10	2.00	2.50	16	20	260	326
1/2	8	2.00	2.00	16	16	398	398
1/3	5	1.00	1.25	8	10	285	356
1/4	4	1.00	1.00	8	8	332	332

TABLE III
SYSTEM LTC WHEN ADAPTIVE LDPC CODES ARE USED AND THE ACHIEVED IMPROVEMENT FACTOR

Maximum Iterarions	**LTC [PB]**		**Improvement Factor**	
	M1	**M2**	**M1**	**M2**
10	574	607	1.73 - 2.62	1.83 - 2.83
20	621	667	1.69 - 2.56	1.82 - 2.39
30	635	685	1.62 - 2.46	1.74 - 2.46
50	641	694	1.61 - 2.49	1.74 - 2.49

TABLE IV
SYSTEM LIFETIME FOR SUSTAINED DATA RATES

Code Rate	**Lifetime [Years]**	**Lifetime [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

The limit in the number of decoder's iterations does not severely affect the lifetime capacity (10%-15%), but it affects I/O performance throughout the lifetime of the NVM-based storage system.

The change of the system's capacity during the aging of its devices is shown in Fig. 5. The lifetime capacity is the sum of the total shaded area. An inevitable drawback of the adaptive code rate scheme is that the user capacity decreases as the number of PE cycles increases. Although this is an undesirable effect, it has to be mentioned that the system's

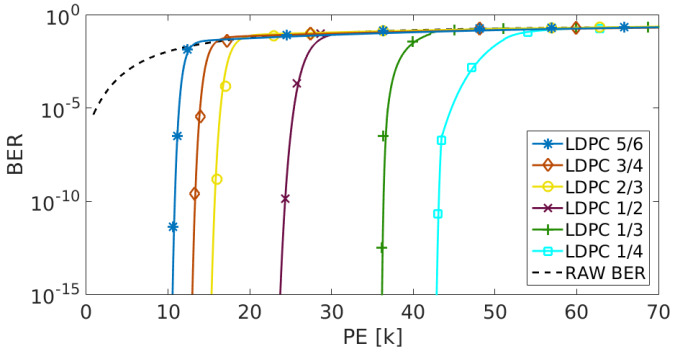

Fig. 3. Raw and User BER versus PE cycling for different LDPC rates

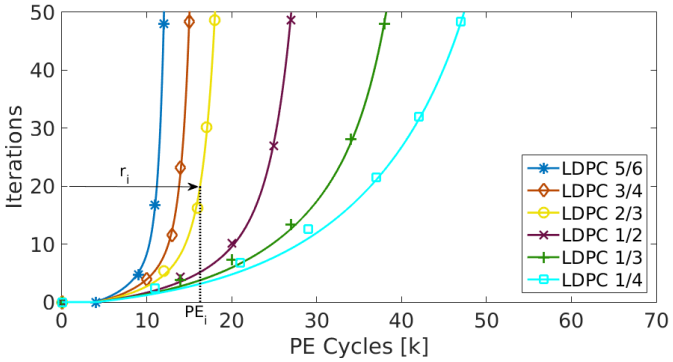

Fig. 4. Number of decoding iterations per LDPC rate versus PE cycles

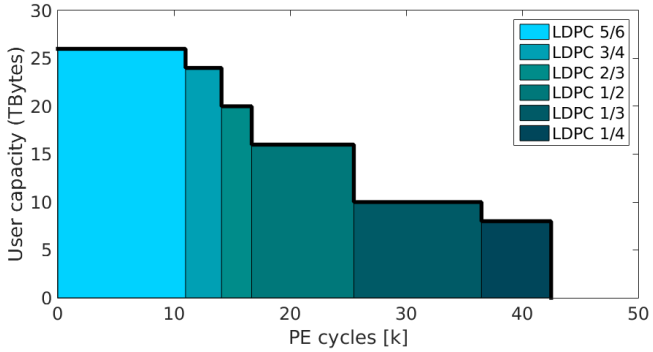

Fig. 5. Evolution of User Capacity when adaptive code rates are used.

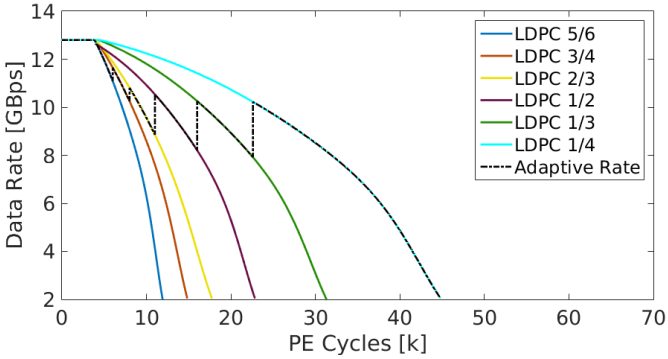

Fig. 6. Evolution of I/O Rate versus PE cycles for various LDPC codes.

reliability remains the same throughout its lifetime, while its lifetime and LTC are much higher compared to any fixed rate configuration. It is obvious that it is preferable to experience less storage capacity than having a fully collapsed system.

B. I/O Performance

In addition to LTC, the use of multiple LDPC codes affects the storage system's I/O performance. As shown in Fig. 6 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. Conclusions

In this paper, we presented the architecture and the performance of a NVM-based storage system that uses multiple, adaptive to the aging conditions, LDPC code rates. The proposed architecture succeeds to almost double the system's lifetime capacity compared to fixed LDPC rate approaches, provides a guaranteed reliability, keeps the total implementation complexity relatively low and achieves high performance. The above storage system architecture is mainly targeting enterprise applications.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

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

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

[7] 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.

[8] 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, 2011, pp. 2-10.

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

[10] 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.

[11] S. Korkotsides, G. Bikas, E. Eftaxiadis and T. Antonakopoulos, *'BER analysis of MLC NAND Flash memories based on an asymmetric PAM model"*, in 6th International Symposium on Communications, Control and Signal Processing (ISCCSP) 2014, pp. 558-561.