

# On Using the Cyclically-Coupled QC-LDPC Codes in Future SSDs

Qing Lu<sup>2</sup>, Chiu-Wing Sham<sup>1,2</sup> and Francis C. M. Lau<sup>1,2</sup>

<sup>1</sup> The Hong Kong Polytechnic University Shenzhen Research Institute, Shenzhen, China

<sup>2</sup> The Hong Kong Polytechnic University, Hong Kong

**Abstract**—As the flash memory continues its capacity scaling and correspondingly decreases its reliability, a technology upgrade regarding the error-correction engine in state-of-art solid-state drives (SSDs) is intensely expected. Due to their limit-approaching decoding ability, low-density parity-check (LDPC) codes are seen as one of the most promising substitute for the traditional BCH codes, though implementation barriers remain to degrade their performance. In our recent work, a co-design of LDPC block codes and their decoder architecture are developed and found suitable to apply to address these barriers with an overall excellence in error rate, complexity as well as throughput. Four codes of 4 KB and 4/5 rate are proposed and their FPGA-based implementations are conducted. It is shown that the decoders reach 1.47 Gb/s throughput at 100 MHz clock rates, and their complexity are estimated to be 1 million gates with 1 Mb memory.

## I. INTRODUCTION

A recent trend that has revolutionized the memory system of modern computing devices is the advent of solid-state drives (SSDs). The flash-based SSDs feature a striking advantage over their traditional counterparts such as hard disks, in the access time of read/write operations rendering more favorable bandwidth and latency. In addition, the growth of their popularity also owes as much to the improved performance in energy efficiency, shock resilience and storage reliability, etc. However, some of these merits are at stake because the technology scaling is always pursued to lower the cost for unit storage space and hence more and more problems have arisen.

One of the largest problems to the vendors' concern is the aggravated susceptibility to noise that would deteriorate the storage reliability [1], [2]. A practicable and efficient method is to modify the controller that is tasked to code/decode the stored information when it is written/read to eliminate errors. For all the commercial SSDs being used today, BCH code is adopted to fulfill this task. Considering its limited error-correction capability, it is acknowledged that low-density parity-check (LDPC) codes should be the most promising candidate for replacement [3].

LDPC codes were first invented by Gallager [4] in 1962 and widely studied and applied in the past two decades. Their credits earned in communication community are mainly due to the best limit-approaching performance which is also the attraction to memory industry [5]. Apart from decoding ability, the growing popularity of LDPC codes is the result of the advancement of their implementations [6]–[10]. But, although the strength of LDPC codes seem to reach a consent and their implementations have made substantial achievements, issues on their applications in SSDs are still challenging. First, the

error-correction capacity is only achievable with iterative soft decoding that is substantially more time-consuming than hard decoding adopted for BCH codes. Second, the encoder/decoder complexity should be restricted such that the holistic cost can be offset. Third, although LDPC codes of various sizes have been standardized for many communication systems, new designs are required for an optimal adaptation to the configurations of target devices.

In our previous work, a structured Cyclically-Coupled Quasi-cyclic LDPC (CC-QC-LDPC) decoder was designed and evaluated to demonstrate its overall excellence for error-correction implementation. We found this design also a match as a solution for the demanding reliability-enhancing problem in SSDs because it features nonexistent or extremely low error floor, high throughput and economical complexity as the same time. Using the original frame, new codes are proposed, dedicated to fitting the size of 4 KB as one SSD page. Decoders are implemented accordingly using field-programmable gate arrays (FPGAs) to compare their performance. The result suggests that the CC-QC-LDPC codes and decoders are competitive candidates for the ECC solution for future SSDs.

## II. BACKGROUND

### A. Solid-State Drive

A typical solid-state drive architecture is shown in Fig. 1. Most manufacturers use NAND flash packages as the main memory fabricate. Between the raw data and I/O interface, an embedded processing component named controller that executes several dispatching and assembling functions plays an predominant role for the overall performance of the SSD. Specifically, it fulfills the error-correction functions at each read/write operation so a decoder module should be embedded within and that is where our design has been focused.

### B. QC-LDPC Codes

QC-LDPC codes are the sub-class of LDPC codes with quasi-cyclic characteristics. In general, their parity-check matrix (PCM) has a representation shown in (1) where each sub-block  $\mathbf{I}$  is a circulant permutation matrix formed by cyclicly shifting the columns of an identity matrix to the right by  $\mathbf{I}^{a_{j,l}}$  ( $0 \leq a_{j,l} \leq z-1$ ) times where  $z \times z$  is the size of the circulant permutation matrix [11].

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}^{a_{1,1}} & \mathbf{I}^{a_{1,2}} & \dots & \mathbf{I}^{a_{1,L}} \\ \mathbf{I}^{a_{2,1}} & \mathbf{I}^{a_{2,2}} & \dots & \mathbf{I}^{a_{2,L}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}^{a_{J,1}} & \mathbf{I}^{a_{J,2}} & \dots & \mathbf{I}^{a_{J,L}} \end{bmatrix}, \quad (1)$$

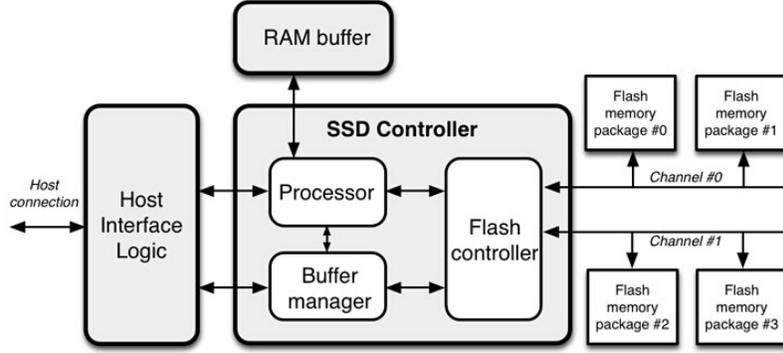


Fig. 1: Architecture of a solid-state drive.

The QC structure endows the decoding with a chance of parallelism, which can significantly increase decoding throughput.

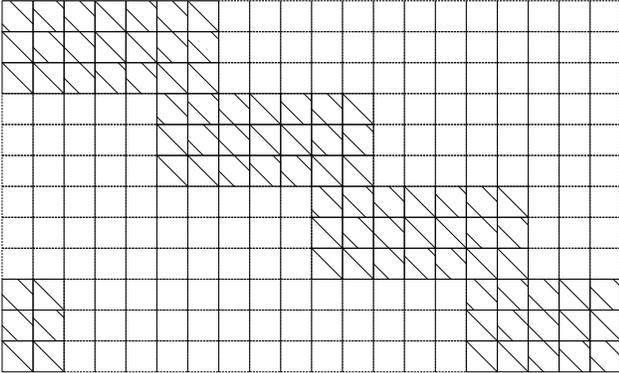


Fig. 2: A sketch of CC-QC-LDPC code structure.

Based on the structure in (1), we developed the cyclically-coupled code layout to relate consecutive codewords by head and tail. A sketch of this layout with is shown in Fig. 2. In this sketch, there are totally four sub-codes whose PCM are partially overlapped with one another. At a cost of lower rate, more variables are constrained very efficiently such that the decoder should be much less complex than those of ordinary codes with the same length. On the other hand, the decoding ability should be significantly stronger than one sub-code. Note that for each variable node in the overlapping region, 6 check nodes are connected to it. Besides, the decoding of all the sub-codes can be performed concurrently, resulting further improvement in throughput.

### C. Decoder Architecture

Following the structure of its PCM, a composite decoder architecture is designed by assembling dedicated QC-LDPC sub-decoders with cyclic interconnect (Fig. 3). For flexibility in the number of components, we use three consecutive sub-decoders to generalize its spirit. Each sub-decoder includes a layered decoder as the main processor, RAMs for soft-message storage and a configurable routing network to bridge

them. This decoder performs a belief-propagation process of layered decoding where messages shuffling between variables and checks are iteratively updated in a row-wised order [12]. Memory blocks are divided into two partitions according to their contents either belonging to a specific sub-code or interchanging information between two sub-codes. With this architecture, the hardware could be both efficient and flexible since each sub-decoder is configurable for reuse by multiple sub-codes if the operations are properly scheduled. The detailed design and operation of the decoder can be found in [13], [14].

## III. IMPLEMENTATION

Operations in most of contemporary SSDs are based on the pages each of size 4 KBytes. Accordingly, we have developed four codes of length 40960 bits and 4/5 rate for the LDPC-in-SSD scheme. If we denote  $K$  as the number of sub-codes and the sub-matrix size is  $z \times z$  for each sub-code, the specifications for all these codes are as follows: (a)  $K = 2$ ,  $z = 1024$ , girth=8; (b)  $K = 4$ ,  $z = 512$ , girth=8; (c)  $K = 4$ ,  $z = 512$ , girth=6; and (d)  $K = 8$ ,  $z = 256$ , girth=6. The other common parameters are  $J = 4$  block rows,  $L = 24$  block columns and  $W = 4$  of them on both sides of a sub-code overlapped with its neighboring sub-codes.

We have implemented the decoders for all these codes using a field-programmable gate array and the implementation details are shown in Table. I. It shows that the hardware sizes are basically identical for all the decoders, which is estimated to be one million gates with 1 Mbits memory. This is because we have managed to adopt the same parallelism such as the same throughput. Therefore, all the decoders can reach up to 1.47 Gbps throughput when they run at 100 MHz clock rates. Note that the clock rate here has been normalized for evaluation while our FPGA-based prototype is able to operate much faster.

Furthermore, we have run simulations to obtain their error performance under AWGN channels as illustrated in Fig. 4. In order to model this channel, a straightforward architecture is implemented to satisfy the throughput requirement [15]. Assuming a BPSK communication, the channel messages are set to be  $2y/\sigma^2$  where  $y$  is the received value and  $\sigma^2$  is

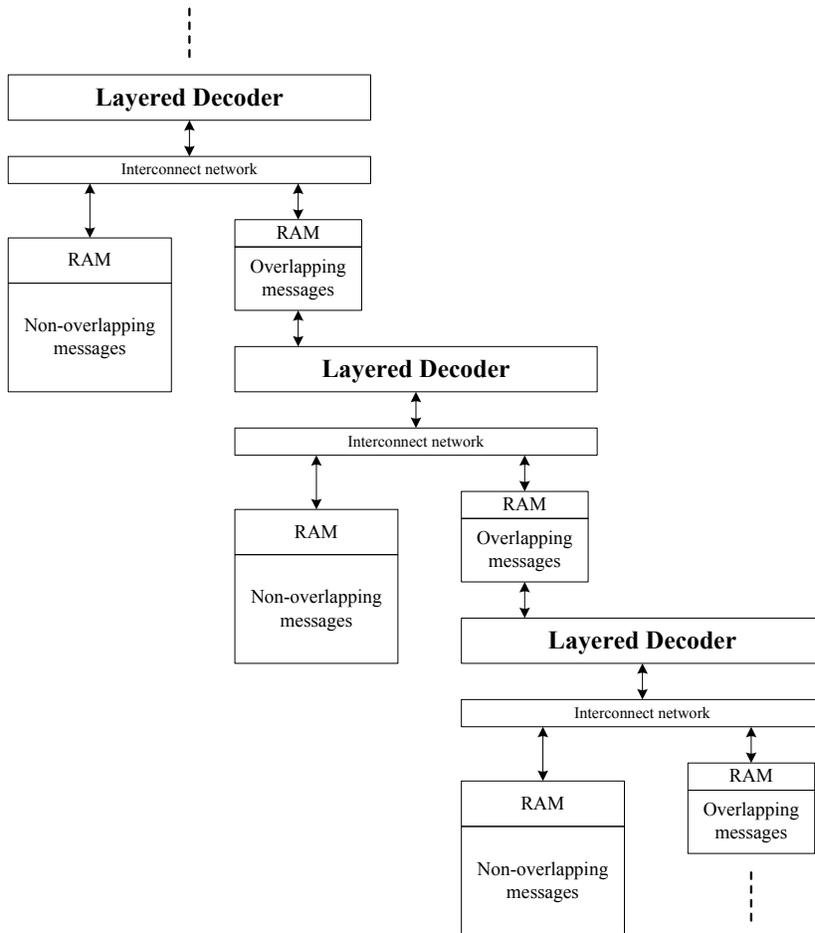


Fig. 3: The architecture of CC-QC-LDPC decoder.

TABLE I: The implementation information of four CC-QC-LDPC decoders with 4-bit quantization and 10-iteration decoding.

| Code               | a         | b         | c         | d         |
|--------------------|-----------|-----------|-----------|-----------|
| parallelism degree | 16        | 8         | 8         | 4         |
| No. ALUTs          | 58,306    | 58,382    | 58,349    | 58,366    |
| Registers          | 37,528    | 37,528    | 37,528    | 37,528    |
| Memory bits        | 1,015,808 | 1,015,808 | 1,015,808 | 1,015,808 |
| Clock rate         | 100 MHz   | 100 MHz   | 100 MHz   | 100 MHz   |
| Throughput         | 1.47 Gbps | 1.47 Gbps | 1.47 Gbps | 1.47 Gbps |

the power of noise. By pre-calculation, the quantized channel messages can be generated using uniform distribution samples and an integrated look-up table that follow a corresponding distribution to the signal-to-noise ratio (SNR).

From Fig. 4, it is observed that the bit-error-rates (BERs) of the codes a, b, and c are so close that the difference among them are less than 0.02 dB at the  $10^{-10}$  level. For all of them, no error floors are observed above  $10^{-11}$ . However, code d demonstrates an inferior error rate than the others with 0.1 dB degradations at the BERs between  $10^{-8}$  and  $10^{-9}$ . Given the implementation information, the other three codes are evidently better choices to be applied and we could take advantage the scalability provided by the decoder to seek better performance.

#### IV. ACKNOWLEDGEMENT

The work described in this paper was supported by the RGC of the Hong Kong SAR, China (Project No. PolyU 152088/15E) and by NSF of China (Grant No. 61372095).

#### V. CONCLUSION AND FUTURE WORK

The implementation and simulation results suggest that our decoder has an overall excellence in error rate, complexity and throughput. As a result, the CC-QC-LDPC codes and decoder have strong potentials for the error-correction solution of future SSDs. To make this proposal applicable, however, several more tasks still need to be accomplished. First, more time shall be spent conducting more simulations to explore the error performances in lower levels, say  $10^{-14}$  and below. Second, as suggested by the simulation results, there is still room to improve the decoding capabilities without hardware degradation by redesigning more powerful codes. Third, different models for the SSD data-sensing channel should also be studied to examine the decoders' all-sided performance.

#### REFERENCES

- [1] L. M. Grupp, J. D. Davis, and S. Swanson, "The bleak future of NAND flash memory," in *Proceedings of the 10th USENIX Conference on File and Storage Technologies*, ser. FAST'12. Berkeley, CA,

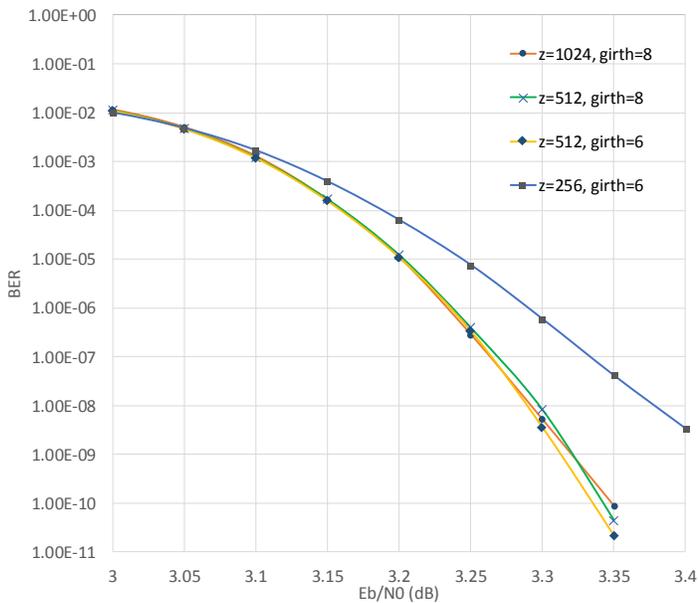


Fig. 4: The bit error rates of the implemented CC-QC-LDPC decoders. The simulation assumes AWGN channel, BPSK modulation and 4-bit quantized soft information.

in Israel, 2004. *Proceedings. 2004 23rd IEEE Convention of*, Sept 2004, pp. 223–226.

- [13] Q. Lu, J. Fan, C. W. Sham, W. M. Tam, and F. C. M. Lau, “A 3.0 Gb/s throughput hardware-efficient decoder for cyclically-coupled QC-LDPC codes,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 1, pp. 134–145, Jan 2016.
- [14] Q. Lu, Z. Shen, C. W. Sham, and F. C. M. Lau, “A parallel-routing network for reliability inferences of single-parity-check decoder,” in *Advanced Technologies for Communications (ATC), 2015 International Conference on*, Oct 2015, pp. 127–132.
- [15] Q. Lu, J. Fan, C. W. Sham, and F. C. M. Lau, “A high throughput Gaussian noise generator,” in *Circuits and Systems (APCCAS), 2014 IEEE Asia Pacific Conference on*, Nov 2014, pp. 117–120.

USA: USENIX Association, 2012, pp. 2–2. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2208461.2208463>

- [2] K. Zhao, W. Zhao, H. Sun, X. Zhang, N. Zheng, and T. Zhang, “LDPC-in-SSD: making advanced error correction codes work effectively in solid state drives,” in *Presented as part of the 11th USENIX Conference on File and Storage Technologies (FAST 13)*. San Jose, CA: USENIX, 2013, pp. 243–256. [Online]. Available: <https://www.usenix.org/conference/fast13/technical-sessions/presentation/zhao>
- [3] X. Hu, “LDPC codes for flash channel,” *Proc. Flash Memory Summit*, 2012.
- [4] R. Gallager, “Low-density parity-check codes,” *IRE Tans. Inf. Theory*, vol. 7, pp. 21–28, 1962.
- [5] E. Yeo, “An LDPC-enabled flash controller in 40 nm CMOS,” *Proc. Flash Memory Summit*, 2012.
- [6] C. W. Sham, X. Chen, W. M. Tam, Y. Zhao, and F. C. M. Lau, “A layered QC-LDPC decoder architecture for high speed communication system,” in *Circuits and Systems (APCCAS), 2012 IEEE Asia Pacific Conference on*, Dec 2012, pp. 475–478.
- [7] J. Zhao, F. Zarkeshvari, and A. H. Banihashemi, “On implementation of min-sum algorithm and its modifications for decoding low-density parity-check (LDPC) codes,” *IEEE Transactions on Communications*, vol. 53, no. 4, pp. 549–554, April 2005.
- [8] Z. Wang and Z. Cui, “Low-complexity high-speed decoder design for quasi-cyclic LDPC codes,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 15, no. 1, pp. 104–114, Jan. 2007. [Online]. Available: <http://dx.doi.org/10.1109/TVLSI.2007.891098>
- [9] Y. Dai, N. Chen, and Z. Yan, “Memory efficient decoder architectures for quasi-cyclic LDPC codes,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 9, pp. 2898–2911, Oct 2008.
- [10] X. Chen, J. Kang, S. Lin, and V. Akella, “Memory system optimization for FPGA-based implementation of quasi-cyclic LDPC codes decoders,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 1, pp. 98–111, Jan 2011.
- [11] M. P. C. Fossorier, “Quasicyclic low-density parity-check codes from circulant permutation matrices,” *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1788–1793, Aug 2004.
- [12] E. Sharon, S. Litsyn, and J. Goldberger, “An efficient message-passing schedule for LDPC decoding,” in *Electrical and Electronics Engineers*