BCH Code Selection and Iterative Decoding for BCH and LDPC Concatenated Coding System

Pin-Han Chen[†], Jian-Jia Weng[†], Chung-Hsuan Wang, and Po-Ning Chen

***Abstract*—In this letter, we consider a concatenated BCH and QC-LDPC coding system for potential use of data protection on flash memory. Two issues are studied, and strategies to resolve them are proposed. First, in order to guarantee that the concatenated coding system is free from undesired error floor, we propose a strategy to select the outer BCH codes according to the error patterns of inner QC-LDPC code. We next present an iterative decoding algorithm between inner QC-LDPC and outer BCH codes to alleviate the performance degradation in the waterfall region due to code-concatenation rate loss. The two proposals jointly provide a feasible design for the concatenated BCH and QC-LDPC coding system. Simulations to verify the performance of the proposed concatenated coding system design are given at the end.**

***Index Terms*—Concatenated coding, iterative decoding, flash memory.**

I. Introduction

Ensuring data reliability on flash memory is an important research issue. For conventional single-level cell flash memory, BCH codes are widely employed for data protection [1]–[4]. However, when being advanced to multi-level cell flash memory, errors of data access turn out to be more likely to occur because of the reduction of margin between state levels. As such, BCH codes may no longer be adequate to meet the error requirement of multi-level cell flash memory. Industries thus start to search for new error correcting codes, such as quasi-cyclic low-density parity-check (QC-LDPC) codes, to protect data in next-generation flash memory.

Among the existing coding techniques, serial concatenation of codes has proven to be a powerful scheme that can dramatically increase the error correcting capability of a resulting system. From the aspect of technology migration, a favorable concatenated coding scheme for next-generation flash memory is perhaps the combination of BCH and QC-LDPC codes. Two advantages of this concatenation can be immediately asserted. First, the existing hardware implementation of BCH encoders and decoders in flash memory industries can be retained. In addition, the special parity-check matrix structure of QC-LDPC codes makes fast encoding as well as low-complexity decoding feasible [5]–[7].

When QC-LDPC codes are decoded by the sum-product algorithm (SPA) [8], a "waterfall"-like sudden drop of the bit error rate (BER) can be commonly observed at low-to-moderate signal-to-noise ratios (SNRs). However, when the SNR further increases, the error performance curve will become flattened, resulting in the so-called "error floor" phenomenon. The concatenation of QC-LDPC codes with properly selected BCH codes, as studied in this letter, is then intended to eliminate this undesired error floor. After settling the error floor by concatenating the BCH codes, the rate loss due to code concatenation will inevitably cause a performance degradation in the waterfall region. We then propose an iterative decoding strategy between the inner QC-LDPC and outer BCH codes to neutralize this degradation.

Specifically, we propose to examine the error patterns of a given QC-LDPC code and identify the dominant error pattern among them. The BCH codes that are going to concatenate with the QC-LDPC code are then selected from those having sufficient error correcting capability to eliminate the dominate error pattern. Since a coding system that is suitable for low cost hardware implementation is more favorable in flash memory, an iterative decoding strategy to neutralize the performance degradation due to code rate loss is better to retain the use of existing hard-output hardware design of the BCH decoder, rather than re-designing a soft-output one. For this reason, we propose to simply feedback those successfully decoding results of outer BCH codes to the inner QC-LDPC code decoder and use simulations to confirm that this simple strategy can compensate the unwanted performance degradation without introducing additional complexity in decoder hardware design.

The rest of the paper is organized as follows. Section II briefs the necessary background. Section III presents the proposed selection method for BCH codes according to the dominant error pattern of the QC-LDPC code, and also the iterative decoding strategy we proposed. Section IV summarizes the simulation results, and Section V concludes this letter.

II. Background

Consider a concatenated coding system comprised of β identical BCH codes as the outer codes and a QC-LDPC code as the inner code. For both BCH and QC-LDPC codes, systematic encoders are used. The information bits are divided into β disjoint groups, where the bits in each group are encoded by their respective BCH encoder. The resulting β BCH codewords are then aggregated and fed into the inner QC-LDPC encoder.[1] A typical decoding procedure for such a concatenated coding system is described as follows. First,

Manuscript received January 17, 2013. The associate editor coordinating the review of this letter and approving it for publication was S. Yousefi.

The authors are with the Dept. of Electrical and Computer Engineering, Nat'l Chiao Tung University, Taiwan 300, ROC (e-mail: {y90633.cm99g, chaselweng.cm95g} @g2.nctu.edu.tw; chwang @mail.nctu.edu.tw; poning @ faculty.nctu.edu.tw).

[†]These authors contributed equally to this work.

Digital Object Identifier 10.1109/LCOMM.2013.031913.130142

[1]When the total length of β BCH codewords is shorter than that of a QC-LDPC codeword, zeros will be appended. Notably, the appended zeros do not need to be stored in flash memory due to the systematic nature of inner codes.

decode directly those y_i's that correspond to the "systematic portion" of an inner QC-LDPC codeword by β BCH decoders, where y_i is the ith noise-contaminated readout from flash memory. If this direct decoding results in β legitimate BCH codewords, stop the decoding procedure; otherwise, launch the QC-LDPC decoder for the entire raw readouts, and re-do the BCH decoding, where decoding results will be forced out using the systematic nature of the adopted code.

A. Review of the LDPC Decoder

A useful graphical representation of an LDPC code with the parity-check matrix $\boldsymbol{H}$ is the Tanner graph [9]. Let v_i and $\mathcal{M}(i)$ be the ith variable node and the set of check nodes connecting to v_i in the Tanner graph, respectively. Similarly, let c_j and $\mathcal{N}(j)$, respectively, be the jth check node and the set of variable nodes connecting to c_j. Denote by R_i the reliability of the ith readout, by $L_{i,j}$ the extrinsic information passing from variable node v_i to check node c_j, and by $E_{i,j}$ the extrinsic information passing from check node c_j to variable node v_i. With these notations, we reproduce the SPA with respect to additive white Gaussian noises below.

The Sum-Product Algorithm (SPA)

Step 0. **Initialization**: *For all i and j, initialize $E_{i,j} = 0$. For all i, assign the value of R_i as $R_i \triangleq 2y_i/\sigma^2$, where σ^2 stands for the noise variance.*

Step 1. **Bit to Check Message Update**: *For all i, and for $j \in \mathcal{M}(i)$ for the given i, compute*

$$L_{i,j} = \sum_{j' \in \mathcal{M}(i), j' \neq j} E_{i,j'} + R_i$$

Step 2. **Check to Bit Message Update**: *For all j, and for $i \in \mathcal{N}(j)$ for the given j, renew*

$$E_{i,j} = \ln\left(\frac{1 + \prod_{i' \in \mathcal{N}(j), i' \neq i} \tanh\left(\frac{L_{i',j}}{2}\right)}{1 - \prod_{i' \in \mathcal{N}(j), i' \neq i} \tanh\left(\frac{L_{i',j}}{2}\right)}\right) \quad (1)$$

Step 3. **Codeword Test**: *Assign the value of D_i according to*

$$D_i = \sum_{j \in \mathcal{M}(i)} E_{i,j} + R_i$$

and determine the hard-decision output of the ith bit by

$$\hat{z}_i = \begin{cases} 1 & \text{if } D_i \leq 0 \\ 0 & \text{if } D_i > 0 \end{cases}$$

If $\boldsymbol{H}\hat{\boldsymbol{z}}^T = 0$, where "$T$" denotes the matrix transpose operation, output $\hat{\boldsymbol{z}}$ (as the decoded result) and stop the algorithm; else, go to Step 1 unless the maximum number of iterations is reached.

B. Review of Trapping Sets in Error Floor Region

As aforementioned, the decoding of LDPC codes by the SPA commonly results in a sudden drop of BERs at low-to-moderate SNRs; however, the BER may stop decreasing when SNR increases beyond a certain value. Such an error floor phenomenon, as having been pointed out by many researchers, is mainly due to the so-called "trapping sets" [10].

By definition, a sub-graph of a Tanner graph is an (a, b) *trapping set* if the sub-graph contains a variable nodes and b odd-degree neighboring check nodes that are connected to these variable nodes. Apparently, only these b check nodes can feed in correct messages to the a variable nodes in the next SPA iteration. Moreover, an (a, b) trapping set is a *dominant trapping set* if it has the smallest b for a given a. Studies in the literature then indicate that the dominant trapping sets are the main errors occurring in the error floor region [11].

III. BCH Code Selection and Iterative Decoding for the Concatenated Coding System

In this section, we introduce how the outer BCH codes are selected in our concatenated coding scheme. In addition, a decision-feedback iterative decoding strategy between the inner BCH and outer QC-LDPC decoders is presented.

Based on the observation in Section II-B, we can infer that if the outer BCH codes selected can correct the errors caused by the dominant trapping sets of the inner QC-LDPC code, the undesired error floor could be removed. It however may be too intensive and also challenging to identify a precise BCH code for a specific dominant trapping set pattern. Instead, we propose to simply identify the parameters (a, b) of the dominant trapping sets and select the BCH codes according to these parameters. Specifically, let a^* be the maximum a of (a, b) dominant trapping sets found by computer search. Then a t-error correctable BCH code with $t \geq a^*$ will be used in our concatenated coding system. By this choice, even if the uncorrected errors from the inner QC-LDPC decoder simultaneously occur in one single BCH coded block, they can be corrected by the outer BCH decoder.

After resolving the error floor issue, we next resolve the performance degradation in the waterfall region due to code-concatenation rate loss. Experiments not included in this letter show that less performance drop results in the waterfall region if outer BCH codes with a higher rate are used; however, the underlying restriction of $t \geq a^*$, where t is the error correcting capability of the outer BCH codes, has placed an upper limit on the attainable code rate. A solution proposed in this letter is to do iterative decoding between the inner and outer codes.

Regarding the input sequence to BCH decoders, we let $\mathcal{S}$ be the set of indices of those bits that can be decoded to valid codewords, where the indices are numbered from the QC-LDPC code's aspect. Notably, the direct correspondences between the input bits of the BCH encoders and the resultant QC-LDPC codeword are well defined since both codes are systematic. For convenience, the iteration between the inner and outer codes is referred to as the *outer iteration* in the sequel, as contrast to the *inner iteration* performed within the QC-LDPC decoder. We then describe the iterative decoding that we propose in the following.

After reading out the raw data from flash memory, the BCH decoders first attempt to decode the systematic portion corresponding to the QC-LDPC codeword. The decoding procedure stops directly, when all the BCH decoders succeed in their decoding; otherwise, the inner QC-LDPC decoder is activated, where the typical sum-product algorithm (cf. Sec. II-A) is executed. If the outputs obtained from the QC-LDPC decoder, again, cannot be successfully decoded by some of the BCH decoders, the following algorithm will be used instead of the typical SPA. The above iterative decoding procedure will continue until either the number of BCH decoders that succeed in their decoding remains the same as that in the previous outer iteration, or the maximum number of outer iterations allowable is reached. In notations, let $\hat{\boldsymbol{d}} = [\hat{d}_1, \hat{d}_2, \hat{d}_3, \ldots]$ and $\hat{\boldsymbol{z}} = [\hat{z}_1, \hat{z}_2, \hat{z}_3, \ldots]$ be respectively the decoded results of the BCH and QC-LDPC decoders.

The Refined SPA for Iterative Decoding

Step 0′. **Initilization**: *For all i and all j, initialize $E_{i,j} = 0$. For all i, assign the value of R_i as*

$$R_i \triangleq \begin{cases} 2(1-2\hat{d}_i)/\sigma^2, & \text{if } i \in \mathcal{S} \\ 2y_i/\sigma^2, & \text{if } i \notin \mathcal{S} \end{cases}$$

where $\hat{\boldsymbol{d}} = [\hat{d}_1, \hat{d}_2, \hat{d}_3, \ldots]$ is obtained from the previous outer iteration, and σ^2 stands for the noise variance.

Step 1′. **Bit to Check Message Update**: *For all i, and for $j \in \mathcal{M}(i)$ for the given i, assign*

$$L_{i,j} = \begin{cases} R_i, & \text{if } i \in \mathcal{S} \\ \sum_{j' \in \mathcal{M}(i), j' \neq j} E_{i,j'} + R_i, & \text{if } i \notin \mathcal{S} \end{cases}$$

Step 2′. **Check to Bit Message Update**: *The same as Step 2 in the typical SPA.*

Step 3′. **Codeword Test**: *Assign the value of D_i according to*

$$D_i = \begin{cases} R_i & \text{if } i \in \mathcal{S} \\ \sum_{j \in \mathcal{M}(i)} E_{i,j} + R_i & \text{if } i \notin \mathcal{S} \end{cases}$$

and determine the hard-decision output of the ith bit by

$$\hat{z}_i = \begin{cases} 1 & \text{if } D_i \leq 0 \\ 0 & \text{if } D_i > 0 \end{cases}$$

If $\boldsymbol{H}\hat{\boldsymbol{z}}^T = 0$, output $\hat{\boldsymbol{z}}$ as the decoded result and stop the algorithm; else, go to Step 1 unless the maximum number of iterations is reached.

Recall that $\mathcal{S}$ is the set of indices of those bits that are decoded to valid codewords by the outer BCH decoders in the previous outer iteration. Consequently, what we have modified to the typical SPA is primarily on the value assignment for the messages that belong to $\mathcal{S}$. The basic idea is to fix the R_i's, $L_{i,j}$'s and D_i's values of these reliable bits before re-doing the QC-LDPC decoding. The bits outside $\mathcal{S}$ may then be corrected owing to these fixed reliable information in the current iteration.

It is worth mentioning that a typical iterative decoding between the inner and outer codes of a concatenated coding scheme generally employs soft-decision decoding for both decoders. In our iterative decoding strategy, we however purposefully retain the use of hard-decision BCH decoders. Two implementation merits can then be rendered. First, after reading the raw data from flash memory, the decoding of the "systematic portion" corresponding to the inner QC-LDPC codeword can be performed efficiently by the outer hard-decision BCH decoders. If this direct decoding succeeds for all BCH decoders, the decoding procedure can be stopped without calling for the QC-LDPC decoder. Secondly, if the direct decoding fails for some BCH decoders, adopting the hard-decision BCH decoders can also reduce the complexity of iterative decoding between the inner and outer codes.

IV. Simulation Results

To verify our selection method for outer BCH codes, the $(4590, 3835)$ QC-LDPC code [12] is adopted as the inner code in simulations. Note that this QC-LDPC code has an evident error floor around BER $= 10^{-5}$ due to its dominant trapping sets with $a^* = 4$. Two choices of BCH codes are tested, which are $(255, 223, 4)$ and $(255, 239, 1)$ BCH codes. The former meets the requirement of our selection method with equality (i.e., $t = a^* = 4$), while the latter has an error correcting capability less than $a^* = 4$. Figure 1 then confirms that no apparent error floor is observed for the former selection; however, the error floor remains for the latter choice.

Next, we examine whether the simple iterative decoding we propose can effectively compensate for the performance degradation in the waterfall region due to code-concatenation rate loss. In order to provide more support to our proposal, a different QC-LDPC code [12] is employed in this experiment. Figure 2 then shows that although concatenating $(12700, 12328)$ QC-LDPC code [12] with either $(255, 223, 4)$, or $(511, 475, 4)$, or $(1023, 983, 4)$ BCH codes can eliminate the error floor since their error correcting capabilities are all equal to the a^* parameter of this QC-LDPC code, performance degradation in the waterfall region remains. After the iterative decoding we propose is applied to these three concatenated coding systems, the BERs of all three systems improve and are below the BER of the QC-LDPC code at all simulated SNRs. It should be mentioned that the number of outer iterations performed is at most two in this experiment; hence, this performance improvement is achieved with only a slight increment in decoding complexity.

A side observation from Fig. 2 is that with outer BCH codes of higher rates, less performance drop results in the waterfall region when no iterative decoding is applied. This indicates that the performance degradation in the waterfall region is mainly caused by the rate loss due to code concatenation. Figure 2 also shows that with the proposed iterative decoding between the inner and outer codes, a system with higher code rate is less favorable at low-to-moderate SNRs; however, a higher code rate is still preferred at high SNRs. Similar observations can be made from Fig. 3.

Finally, we compare our concatenating coding system with a conventional flash memory coding design comprised of a single BCH code under similar code rate. Note that the $(12700, 11531)$ BCH code is a shortened code from the $(16383, 15214, 84)$ BCH code [4]. Figure 4 then reveals that the concatenated coding scheme has better performance than that of the $(12700, 11531)$ BCH code, even without implementing the proposed iterative decoding.

V. Conclusion

In this letter, we propose to select the outer BCH codes of a concatenating coding system based on the knowledge of the dominant trapping sets of the inner QC-LDPC code. An iterative decoder for this concatenated coding system is also presented. Simulations validate that the BCH codes selected can result in a concatenating coding system with no

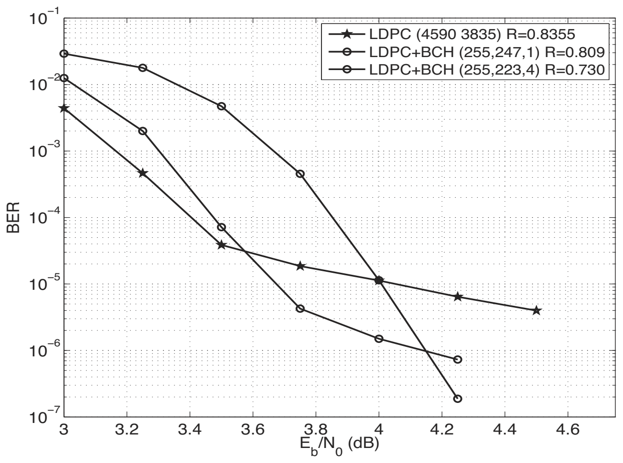

Fig. 1. BERs of the coding systems concatenating the $(4590, 3835)$ QC-LDPC code with different BCH codes. In these simulations, BPSK signaling suffering additive white Gaussian noise (AWGN) is assumed, and the maximum number of inner iterations is 100. No outer iteration is performed. The resulting code rate for each coding system is shown in the legend.

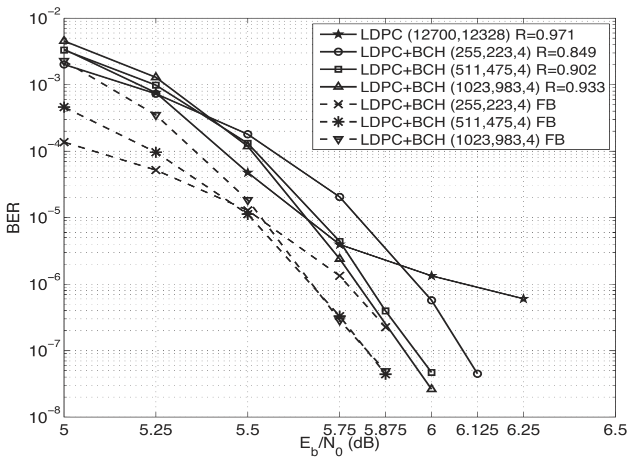

Fig. 2. BERs of the coding systems concatenating the $(12700, 12328)$ QC-LDPC code with different BCH codes. The system settings are the same as in Fig. 1. In the legend, "FB" indicates that the iterative decoding we propose is used.

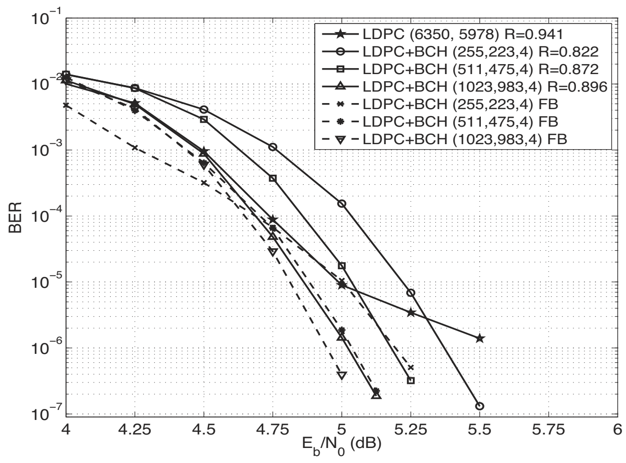

Fig. 3. Redoing Fig. 2 using different codes.

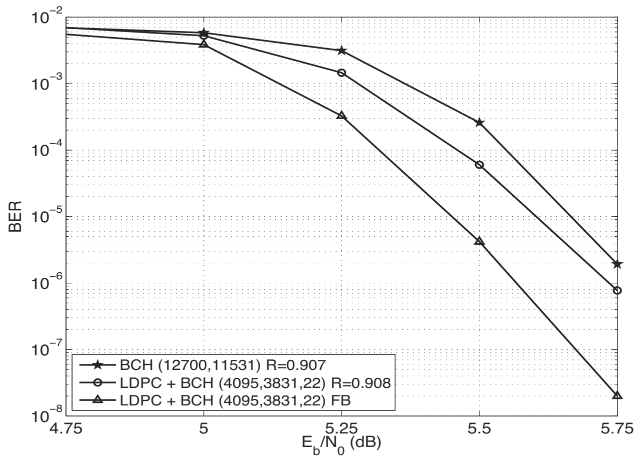

Fig. 4. BERs of the coding system concatenating the $(12700, 12328)$ QC-LDPC code with $(4590, 3831, 22)$ BCH codes. Also shown is the BER of a single $(12700, 11531)$ BCH code.

visible error floor, and the proposed iterative decoding can effectively compensate the performance degradation due to code-concatenation rate loss. In its structure, our proposed system retains the use of the existing hard-output hardware design of the BCH decoders, making it an attractive design for a flash memory system.

References

[1] F. Sun, K. Rose, and T. Zhang, "On the use of strong BCH codes for improving multilevel NAND flash memory storage capacity," in *Proc. 2006 IEEE Workshop on SiPS*.

[2] W. Liu, J. Rho, and W. Sung, "Low-power high-throughput BCH error correction VLSI design for multi-level cell NAND flash memories," in *Proc. 2006 IEEE Workshop on SiPS*, pp. 303–308.

[3] X. Wang *et al.* (2011), "Error correction codes and signal processing in flash memory." Available: http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/

[4] S. Lin and D. J. Costello, *Error Control Coding*, 2nd edition. Prentice Hall, 2004.

[5] L. Lan, L. Zeng, Y. Y. Tai, L. Chen, S. Lin, and K. Abdel-Ghaffar, "Construction of quasi-cyclic LDPC codes for AWGN and binary erasure channels: a finite field approach," *IEEE Trans. Inf. Theory*, vol. 53, no. 7, pp. 2429–2458, Jul. 2007.

[6] L. Chen, L. Zeng, S. Lin, and W. H. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes," *IEEE Trans. Commun.*, vol. 54, no. 1, pp. 71–81, Jan. 2006.

[7] Y. Dai, N. Chen, and Z. Yan, "Memory efficient decoder architectures for quasi-cyclic LDPC codes," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 9, pp. 2898–2911, Oct. 2008.

[8] R. Gallager, "Low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[9] R. M. Tanner, "A recursive approach to low-complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 5, pp. 533–547, Sep. 1981.

[10] T. J. Richardson, "Error floors of LDPC codes," in *Proc. 2003 Allerton Conf. on Commun., Control and Computing*, pp. 1426–1435.

[11] X. Zheng, F. C. M. Lau, and C. K. Tse, "Constructing short-length irregular LDPC codes with low error floor," *IEEE Trans. Commun.*, vol. 58, no. 10, pp. 2823–2834, Oct. 2010.

[12] I. Djurdjevic, J. Xu, K. Abdel-Ghaffar, and S. Lin, "A class of low-density parity-check codes constructed based on Reed-Solomon codes with two information symbols," *IEEE Commun. Lett.*, vol. 7, no. 7, pp. 317–319, July 2003.