

IEEE Standard for Error Correction Coding of Flash Memory Using Low-Density Parity Check Codes

IEEE Computer Society

Sponsored by the
Microprocessor Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

IEEE Std 1890™-2018

IEEE Standard for Error Correction Coding of Flash Memory Using Low-Density Parity Check Codes

Sponsor

Microprocessor Standards Committee
of the
IEEE Computer Society

Approved 27 September 2018

IEEE-SA Standards Board

Abstract: A two-level code construction scheme for non-volatile memories (NVM) that is based on low-density parity-check codes is specified in this standard. This scheme constructs an auxiliary codeword that encodes a subset of bits from the primary packets stored in an NVM memory unit. The auxiliary codeword is decoded only when the detection of at least one of the primary codewords fails. The encoding and decoding techniques for this scheme are presented. The two-level scheme outperforms the traditional one-level method while it requires only a small memory overhead and negligible latency. Moreover, it outperforms the one-level scheme that uses a code that is twice as long in the low raw bit error rate regime.

Keywords: auxiliary codeword, flash memory, IEEE 1890™, low-density parity-check codes, two-level code construction

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2019 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 28 February 2019. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-1-5044-5355-4 STD23440
Print: ISBN 978-1-5044-5356-1 STDPD23440

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.” They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/ipr/disclaimers.html>.

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (“IEEE-SA”) Standards Board. IEEE (“the Institute”) develops its standards through a consensus development process, approved by the American National Standards Institute (“ANSI”), which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed through scientific, academic, and industry-based technical working groups. Volunteers in IEEE working groups are not necessarily members of the Institute and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854 USA

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. A current IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit IEEE Xplore at <http://ieeexplore.ieee.org/> or contact IEEE at the address listed previously. For more information about the IEEE-SA or IEEE's standards development process, visit the IEEE-SA Website at <http://standards.ieee.org>.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: <http://standards.ieee.org/findstds/errata/index.html>. Users are encouraged to check this URL for errata periodically.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at <http://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this IEEE standard was completed, the ECC NVM P1890 Working Group had the following membership:

Kiran Gunnam, *Chair*
Tinoosh Mohsenin, *Secretary*

Behzad Amiri	Zion Kwok	Osso Vahabzadeh
Paul Budnik	Ravi Motwani	Kasra Vakulinia
David Declercq	Scott Nelson	Bane Vasic
Chris Dick	Shiva Planjery	Rick Wesel
Lara Dolecek	Minghai Qin	Rosanna Yee
Erdem Karayer	Paul Siegel	

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Demetrio Bucaneg Jr.	Kiran Gunnam	Osso Vahabzadeh
Keith Chow	Werner Hoelzl	Dmitri Varsanofiev
James Cloos	Ralph Kearfott	Oren Yue
Randall Groves	Ganesh Subramanian	

When the IEEE-SA Standards Board approved this standard on 27 September 2018, it had the following membership:

Jean-Phillipe Faure, *Chair*
Gary Hoffman, *Vice Chair*
John D. Kulick, *Past Chair*
Konstantinos Karachalios, *Secretary*

Ted Burse	Xiaohui Liu	Robby Robson
Guido Hiertz	Kevin Lu	Dorothy Stanley
Christel Hunter	Daleep Mohla	Mehmet Ulema
Joseph Koepfinger*	Andrew Myles	Phil Wennblom
Thomas Koshy	Paul Nikolich	Philip Winston
Hung Ling	Ron Petersen	Howard Wolfman
Dong Liu	Annette Reilly	Jingyi Zhou

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 1890-2018, IEEE Standard for Error Correction Coding of Flash Memory Using Low-Density Parity Check Codes.

This standard proposes a method to achieve higher error correction capability in flash memories using low-density parity-check (LDPC) codes. The proposed techniques can be used with a variety of LDPC encoder and decoder architectures. However, efficient decoder implementations require employing features such as datapath organization, out-of-order processing for Rnew, out-of-order processing, memory organization such as using the same memory for L and Q values and using the same memory for L and P values, value-reuse property, check node unit designs, decoder scheduling features, code construction constraints for efficient hardware implementation and numerous other features, which are equally applicable for binary and non-binary decoders. These features are covered by several issued and pending patents owned by Texas A&M University System (TAMUS) and, as observed by IEEE-SA P1890 Working Group, they provide memory and logic savings up to 75% compared to standard LDPC implementation (see [B10]).¹ These patents are controlled by TexasLDPC Inc. dba Symbyon Systems and can be licensed from them directly. There are no assurances regarding these patents as described in the Letter of Assurance provided to the IEEE Standards Association. The following is the list of the above-referenced issued and pending patents:

- US8418023B2: Low density parity check decoder for irregular LDPC codes
- US8656250B2: Low density parity check decoder for regular LDPC codes
- US8555140B2: Low density parity check decoder for irregular LDPC codes
- US8359522B2: Low density parity check decoder for regular LDPC codes
- US9112530B2: Low density parity check decoder
- US20170093429A1: Low density parity check decoder
- US20150311917A1: Low density parity check decoder
- Provisional Application No. 60/988680
- Provisional Application No. 60/915320

Acknowledgments

The technical document (pages 8–56) reprinted with permission from TexasLDPC Inc., P1890/D2 Draft Standard for Error Correction Coding of Flash Memory Using Low-Density Parity Check Codes, © 2017.

¹The numbers in brackets correspond to those of the bibliography in Annex J.

Contents

1. Overview	9
1.1 Scope	9
1.2 Purpose	9
1.3 Introduction	9
2. Normative references	10
3. Acronyms and abbreviations	10
4. The two-level code construction scheme	10
4.1 Encoding	10
4.2 Decoding	11
5. Decoder Memory Requirements	12
6. Simulation Results	12
Annex A (informative) The shift coefficients of the 1 kB code with rate 0.83 and circulant size 140 simulated in Figure 3	15
Annex B (informative) The shift coefficients of the 2 kB code with rate 0.83 and circulant size 140 simulated in Figure 3	17
Annex C (informative) The shift coefficients of the 1 kB code with rate 0.91 and circulant size 140 simulated in Figure 4	23
Annex D (informative) The shift coefficients of the 2 kB code with rate 0.91 and circulant size 140 simulated in Figure 4	24
Annex E (informative) Bit indexes of the primary codewords encoded by the auxiliary codeword in the simulations of Figure 3 and Figure 4	27
Annex F (informative) The shift coefficients of a 1 kB code with rate 0.88 and circulant size 140	39
Annex G (informative) The shift coefficient of a 2 kB code with rate 0.88 and circulant size 140	41
Annex H (informative) The shift coefficients of a 1 kB code with rate 0.94 and circulant size 140	45
Annex I (informative) The shift coefficients of a 2 kB code with rate 0.94 and circulant size 140	46
Annex J (informative) Bibliography	49

IEEE Standard for Error Correction Coding of Flash Memory Using Low-Density Parity Check Codes

1. Overview

1.1 Scope

This draft standard specifies a method to construct two-level low-density parity-check (LDPC) codes and to utilize them as the error correction coding (ECC) scheme in non-volatile memories (NVM). The encoding and decoding methods as well as the implications on memory and overall system latency are presented. The simulation results comparing the two-level code construction scheme and the traditional one-level scheme, as well as the parity check matrices for several LDPC code rates and lengths are provided.

1.2 Purpose

This draft standard provides a method to utilize error correction coding in non-volatile memories. For this purpose, a two-level code construction method based on LDPC codes along with the decoding method is presented. This scheme outperforms the traditional method, while having slightly higher memory requirements and negligible delay.

1.3 Introduction

Reliable data communications and storage is achieved by employing ECC techniques where desired reliability is determined based on the application. The smallest memory unit in most flash memories is usually used to store several data packets or codewords that are encoded using an ECC technique. For given packet length, the available memory may not be entirely consumed by the stored codewords. For this scenario, the presented two-level code construction scheme constructs an auxiliary codeword using the primary codewords stored in the memory unit that utilizes only the unconsumed memory, which is insufficient to store an entire codeword. This scheme is based on LDPC codes (Gallager [B4], Mackay [B7]) and improves the frame error rate (FER) performance for a given raw bit error rate (RBER) with negligible latency.

In the two-level code construction scheme, the first level comprises encoding the user information bits using T codewords and the second level comprises encoding a subset of each one of the codewords in level one using an auxiliary codeword. However, only the parity bits of the auxiliary codeword are stored and thus, the introduced overhead is small especially for high rate codes.

The decoding is also done in two stages. First, the T primary codewords are decoded. If the detection of at least one of the primary codewords fails, the auxiliary codeword is decoded. Then the extrinsic information obtained from decoding the auxiliary code is provided to the failed packets and these packets are decoded

one more time. This constitutes one global iteration. Since most packets are decoded successfully at high FERs, the latency due to utilizing the auxiliary code to decode the failed codewords is negligible. With a small overhead, the two-level scheme outperforms the traditional scheme where no auxiliary codeword is used. Moreover, it outperforms the traditional scheme that uses a code that is twice as long in the low raw bit error rate (RBER) regime.

2. Normative references

This document does not contain any normative references.

3. Acronyms and abbreviations

BPSK	binary phase-shift keying
ECC	error correction coding
FER	frame error rate
HD	hard decision
kB	kilobyte
LDPC	low-density parity-check
LLR	log-likelihood ratio
NVM	non-volatile memory
QC-LDPC	quasi-cyclic LDPC
RBER	raw bit error rate
SNR	signal-to-noise ratio

4. The two-level code construction scheme

4.1 Encoding

Figure 1 illustrates the encoder block diagram using the two-level code construction scheme. The first level comprises T encoders to encode T information blocks and to generate codewords $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_T$, where T is an arbitrary number. The codewords can be encoded using either the parity check matrices $\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_T$ or the corresponding generator matrices $\mathbf{G}_1, \mathbf{G}_2, \dots, \mathbf{G}_T$ using any of the various techniques to encode LDPC codes (e.g., Gallager [B4], Li et al. [B6], Mackay [B7], and Richardson and Urbanke [B8]). The bits in each codeword $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_T$, shall be interleaved first. Then $k'_1, k'_2, \dots, k'_T$ bits shall be selected from each codeword, respectively, such that $k'_1 + k'_2 + \dots + k'_T = k_{\text{aux}}$, where k_{aux} is the number of bits encoded by the auxiliary encoder. Then these bits shall be encoded using the second level code, which is the auxiliary code. One can use the parity check matrix of the auxiliary code, $\mathbf{H}_{\text{aux}}$ or, equivalently, its generator matrix, $\mathbf{G}_{\text{aux}}$, to obtain the auxiliary codeword $\mathbf{c}_{\text{aux}}$. The auxiliary code may be a systematic code in which case it is represented as $\mathbf{c}_{\text{aux}} = [\mathbf{u}_{\text{aux}}, \mathbf{p}_{\text{aux}}]$, where $\mathbf{u}_{\text{aux}}$ denotes the information bits, which shall be selected from the primary codewords $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_T$. Furthermore, $\mathbf{p}_{\text{aux}} = [c_{a,1}, c_{a,2}, \dots, c_{a,p_a}]$ denotes the parity bits. Since the information bits are already stored, only the generated parity bits, i.e., $\mathbf{p}_{\text{aux}}$, should be stored.

Figure 2 represents the Tanner graph for the two-level code where the check nodes and the variable nodes are illustrated using squares and circles, respectively. Additionally, the upper level Tanner graphs correspond to the codes in the first level and the lower level represents the Tanner graph of the auxiliary code. As it is also evident from this figure, the additional storage overhead corresponds to storing p_a parity bits of the auxiliary codeword.

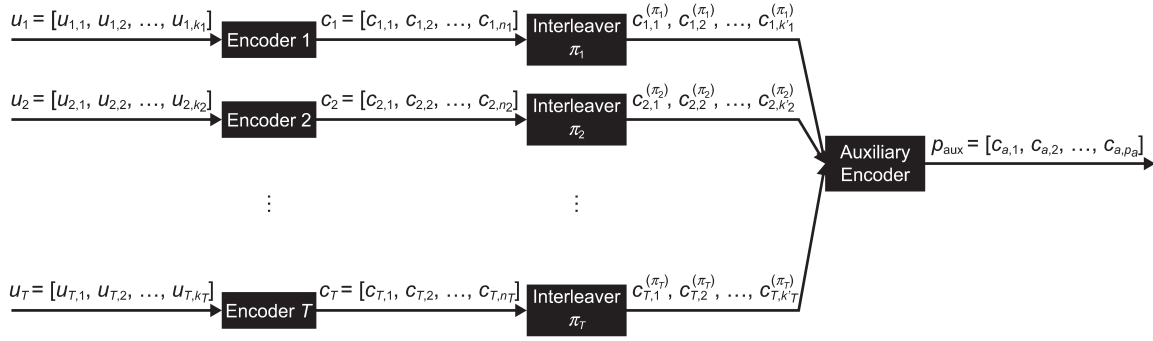


Figure 1—The encoder block diagram for the two-level code construction scheme

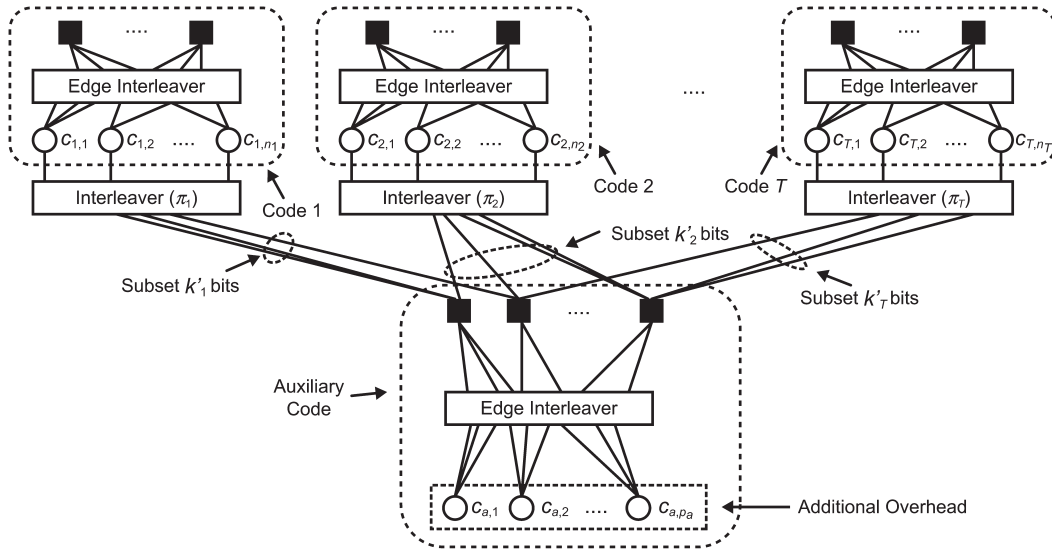


Figure 2—The Tanner graph for the two-level code construction scheme

Remark: As a special case and in the remainder of this work, it is assumed that $\mathbf{H}_1 = \mathbf{H}_2 = \dots = \mathbf{H}_T = \mathbf{H}_{\text{aux}}$. This selection is useful when it is desired to use the same $\mathbf{H}$ matrix to encode and decode the primary codes and the auxiliary code. However, even if the $\mathbf{H}$ matrices are different, a multi-rate decoder circuit can be used to decode all of the codewords as long as the $\mathbf{H}$ matrices have the same circulant size. The only difference would be the scheduling information of the decoder, which depends on each $\mathbf{H}$ matrix. Similarly, a multi-rate encoder can be used to encode all the codes.

4.2 Decoding

The primary goal of employing the auxiliary code is to provide additional information for each failed primary codeword to increase the probability of a successful detection. Thus, the auxiliary codeword shall be decoded only when at least one of the primary codewords in the memory page fails. The codewords can be decoded using various iterative decoding methods and their modifications, such as the sum-product or the min-sum algorithms (e.g., Chen and Fossorier [B1], Fossorier [B3], and Kschischang et al. [B5]).

In the two-level code constructions scheme, the primary and auxiliary code decoders shall exchange information in order to enhance the overall reliability of the messages. One global decoding iteration consists of decoding the auxiliary codeword based on the detection results of the primary codewords followed by decoding the failed codeword based on the detection results of the auxiliary codeword.

Remark: To avoid unnecessary area overhead, one decoder circuitry should be used to decode the primary and auxiliary codewords. In this case, some kind of double buffering can be used to store the necessary decoder messages of the primary and auxiliary codewords. This aspect will be described in more details in [Clause 5](#), where the memory requirements of the two-level code construction scheme will be discussed.

The decoding algorithm is summarized as follows:

Step 1. The T primary codewords in one memory page shall be decoded. Suppose the number of decoding failures are denoted by d_f . If $d_f = 0$, the T primary codewords in the next page shall be decoded; otherwise,

for $m = 1, \dots, d_f$

Step 2. The auxiliary codeword shall be decoded if it has not been decoded successfully in the previous round(s). For decoding the auxiliary codeword, the log-likelihood ratios (LLRs) of the information bits can be computed based on the detection results of the primary codewords and the LLRs of the parity bits can be computed based on the read results of the corresponding parity portion that are stored in the memory page.

Step 3. The LLRs for the codeword bits of the failed primary codeword that are encoded using the auxiliary codeword can be updated based on the decoding results of the auxiliary codeword and provided to the failed codeword's decoder. Then the failed codeword shall be decoded another time.

5. Decoder Memory Requirements

It is useful that some of the memories used in the LDPC decoder are double buffered to make seamless processing of the packets possible. Furthermore, as indicated in [4.2](#), one decoder circuitry should be used to decode the primary codewords and the auxiliary codeword. However, an additional hard decision (HD) memory should be allocated to store the hard decisions of the primary codeword bits that are encoded using the auxiliary packet. Moreover, an HD memory should be used to store the hard-decisions of the failed codewords. Finally, a read-only memory (ROM) should be used to store the connectivity information of the auxiliary codeword and the primary codewords.

To compare the memory requirements of 1 kilobyte (kB) and 2 kB codes, it is assumed that the circulant size, the average variable node degree, and the bit precision of the decoder messages are the same in both cases. Furthermore, it is assumed that $b_{c,2kB} = 2b_{c,1kB}$, $b_{r,2kB} = 2b_{r,1kB}$, and $d_{c,2kB} = 2d_{c,1kB}$, where b_c , b_r , and d_c represent the number of block columns, the number of block rows, and the check node degrees, respectively. These assumptions are reasonable when comparing 1 kB and 2 kB codes. Under these assumptions, the memory requirements of the 2 kB decoder are at least twice as large as the 1 kB decoder.

6. Simulation Results

[Figure 3](#) illustrates the performance results for an LDPC code with codeword length equal to 1 kB (10080 bits) and rate $R = 0.83$. It is assumed that $T = 16$, and thus, the memory page stores 16 primary codewords as well as the parity portion of the auxiliary codeword. It is assumed that $\mathbf{H}_1 = \mathbf{H}_2 = \dots = \mathbf{H}_{16} = \mathbf{H}_{\text{aux}}$ and that the parity-check matrix is a quasi-cyclic LDPC (QC-LDPC) code (Fosserier [\[B2\]](#)) where the circulant size is 140. Moreover, binary phase-shift keying (BPSK) signaling is employed.

It is observed that the auxiliary code improves the performance of the 1 kB LDPC code in all RBERs. For example, the scheme with an auxiliary code achieves an FER equal to 10^{-4} at a channel condition whose RBER is 1.45×10^{-3} worse than the case where no auxiliary code is employed. This is equivalent with 0.08 dB improvement in terms of E_b/N_0 . Furthermore, at RBERs higher than 2.6×10^{-3} , the two-level code construction scheme with codeword length 1 kB outperforms a 2 kB LDPC code that has the same rate. As indicated above, the memory requirements of the scheme using a 2 kB code is more than twice the requirements for the 1 kB scheme with the auxiliary codeword.

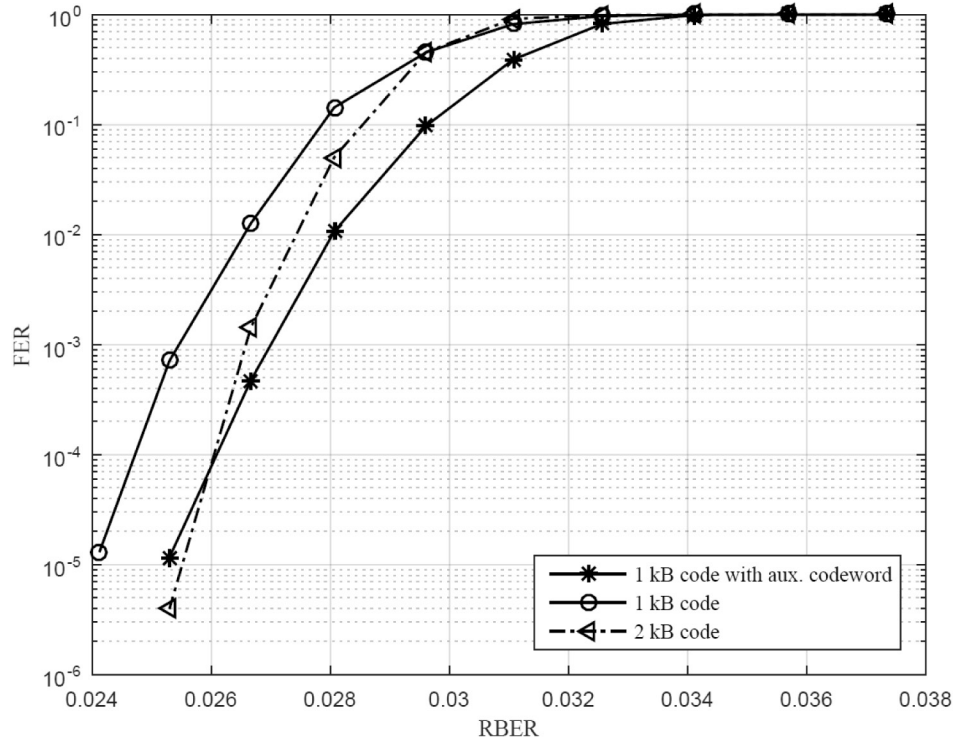


Figure 3—Performance results for codes with codeword rate $R = 0.83$

Figure 4 illustrates the performance results for a 1 kB (9240 bits) LDPC code with rate 0.91. Except for the codeword rate and length, other assumptions made in the previous example are also valid here. Here again, the two-level code outperforms the one-level code in all channel conditions and outperforms the 2 kB long in the low RBER regime.

The latency due to employing an auxiliary codeword, which is only introduced when the decoder fails for at least one primary codeword in the memory unit, is related to five operations. The first operation is to load the hard-decisions related to the memory unit's last primary codeword in the auxiliary codeword's HD memory. The number of clock cycles that is necessary for transferring this information for the last codeword is equal to the number of block columns, i.e., b_c . Therefore, the latency introduced due to this operation is b_c clock cycles per memory unit. It is noted that by the time that the last primary codeword is decoded, the hard-decisions of the previous codewords are already loaded in the auxiliary codeword's HD memory and thus, the only applicable delay is due to the last codeword in the memory unit. However, if the decoder fails for none of the codewords in the unit, loading the hard-decisions of the last packet can be avoided and thus this operation would impose no delays. The second operation that influences the latency is to read the parity bits of the auxiliary codeword from the memory when it is necessary to decode the auxiliary codeword. The corresponding delay depends on the number of parity bits and is smaller for high rate codes compared to low rate codes. The third operation is to decode the auxiliary codeword for which the delay depends on the decoder parallelization and the H matrix structure. The fourth operation is to load the HD detection results corresponding to the failed codeword into the decoder before it is decoded one more time. The HD results may be loaded from the HD memory of the auxiliary codeword or from a queuing memory that is implemented to address statistical buffering. Loading the HD results would take an additional b_c clock cycles. Finally, the last operation is to decode the failed codeword one more time. The latency due to this operation depends on the decoding time of the codeword and is not deterministic.

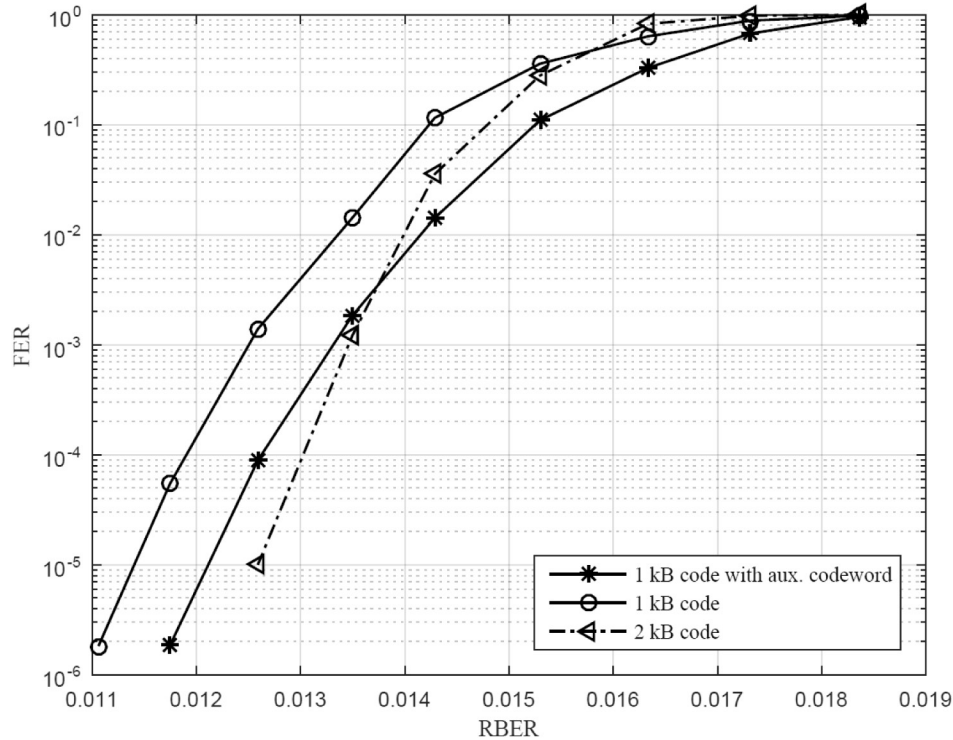


Figure 4—Performance results for codes with codeword rate $R = 0.91$

The latency due to utilizing the two-level code construction scheme as explained above is small in practice. For example, at $\text{FER} = 10^{-4}$, only 1 out of 10000 packets fails to converge on the average. Thus, the latency of the two-level code construction scheme and the case where no auxiliary codeword is used is the same for 99.99% of the packets. For the remaining packets, the latency overhead depends on the codeword rate and length, though the overall latency ratio would be similar for all cases. For example, the 1 kB code simulated in Figure 4 has 66 block columns and thus the first operation requires 66 clock cycles. Additionally, this code has 6 parity blocks where each parity block has 140 bits. Suppose that 35 bits are read from the memory at each clock cycle. Thus, the latency due to reading the parity bits from the memory is $6 \times 140 / 35 = 24$ clock cycles. Finally, this code has 264 non-zero circulants. When the decoder is capable of processing one non-zero circulant per clock, one iteration would require 264 clock cycles. Suppose the average number of iterations at $\text{FER} = 10^{-4}$ is 3 iterations. Hence, decoding the auxiliary codeword introduces a delay equal to 792 clock cycles. Finally, loading the HD results before re-decoding the failed codeword requires 66 clock cycles and the decoding would require another 792 clock cycles. Therefore, the overall delay is equal to $66 + 24 + 792 + 66 + 792 = 1740$ clock cycles and the latency is $1740 / (16 \times 792) = 13.7\%$, which is only the case for 0.01% of the packets. The overall latency is even smaller at lower FERs.

The shift coefficients of the QC-LDPC codes simulated in Figure 3 and Figure 4 are provided in Annex A, Annex B, Annex C, and Annex D, where a ‘-1’ indicates a zero circulant and a ‘0’ indicates that the identity matrix has not been shifted. The remaining values indicate a shift value between 1 and 139. Recall that the circulant size is equal to 140 and thus, every circulant is a 140×140 matrix.

Annex E includes the bit indexes of the primary codewords encoded using the auxiliary codeword in the examples of Figure 3 and Figure 4.

Annex A

(informative)

The shift coefficients of the 1 kB code with rate 0.83 and circulant size 140 simulated in Figure 3

Columns 1 through 13

-1	-1	-1	-1	-1	-1	-1	49	33	78	-1	85	-1
-1	20	0	75	-1	-1	4	-1	-1	-1	-1	-1	-1
46	-1	-1	121	-1	-1	-1	35	-1	-1	65	139	-1
-1	-1	38	-1	77	-1	-1	-1	-1	55	-1	-1	-1
138	-1	112	-1	-1	-1	-1	133	-1	-1	132	-1	22
-1	21	47	-1	93	-1	-1	-1	60	-1	-1	-1	-1
-1	-1	-1	-1	-1	137	78	-1	129	-1	-1	-1	-1
-1	-1	-1	101	-1	-1	-1	63	-1	-1	119	85	56
-1	-1	-1	69	137	118	29	-1	-1	103	-1	-1	45
70	-1	-1	-1	-1	127	66	-1	-1	-1	85	-1	-1
-1	49	-1	-1	26	83	-1	-1	-1	-1	-1	107	60
3	119	-1	-1	-1	-1	-1	-1	66	105	-1	-1	-1

Columns 14 through 26

-1	42	111	-1	-1	-1	111	-1	-1	109	-1	-1	-1
46	-1	-1	-1	-1	9	-1	-1	-1	-1	110	-1	-1
-1	-1	-1	-1	33	-1	96	106	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	113	-1	-1	-1	96	-1	121	14
118	52	-1	-1	110	23	-1	-1	114	-1	-1	57	79
-1	42	41	38	-1	-1	-1	-1	33	45	107	-1	-1
115	101	70	83	-1	-1	-1	-1	93	-1	-1	13	-1
-1	-1	-1	-1	28	-1	19	61	-1	-1	72	-1	114
-1	-1	-1	-1	-1	-1	-1	71	-1	-1	-1	-1	112
-1	-1	-1	100	-1	-1	-1	103	-1	-1	-1	-1	-1
-1	-1	72	-1	-1	-1	-1	-1	117	-1	83	-1	-1
77	-1	-1	2	70	109	40	-1	-1	138	-1	78	-1

Columns 27 through 39

-1	7	-1	-1	-1	-1	-1	139	-1	22	-1	15	-1
-1	133	32	-1	48	-1	75	-1	-1	88	41	-1	126
11	-1	0	53	-1	-1	-1	-1	101	-1	-1	45	-1
116	-1	-1	-1	81	12	-1	-1	-1	-1	-1	-1	25
-1	-1	-1	-1	-1	-1	69	-1	41	-1	6	-1	-1
-1	-1	-1	-1	-1	81	-1	-1	-1	126	-1	-1	-1
-1	10	-1	-1	-1	-1	-1	-1	-1	-1	116	-1	-1
105	-1	103	89	-1	-1	-1	133	101	-1	-1	47	-1
-1	-1	-1	98	69	104	-1	-1	-1	50	-1	-1	-1
38	-1	116	79	-1	-1	123	-1	104	-1	135	34	-1
-1	-1	-1	-1	127	61	-1	48	-1	-1	-1	-1	76
-1	101	-1	-1	-1	-1	85	40	-1	-1	-1	-1	24