A Reduced Complexity Linear QC-LDPC Encoding with Parity Vector Correction Technique

Chanho YOON†, Nonmember, Hoojin LEE††‡, Member, and JoonhyaKANG†††, Nonmember

SUMMARY A new approach for encoding one class of quasi-cyclic low-density parity-check (QC-LDPC) codes is proposed. The proposed encoding method is applicable to parity-check matrices having dual-diagonal parity structure with single column of weight three in the parity generation region. Instead of finding the parity bits directly, the proposed method finds parity bits through vector correction. While the proposed LDPC encoding scheme is readily applicable to matrices defined in the IEEE physical layer standards, the computational complexity of the post processing operation for extraction of correction vector requires less effort than solving the linear equations involved with finding the parity bit as proposed by Myung et al.

1. Introduction

Ever since the rediscovery of low-density parity-check (LDPC) codes [1], making them practical has been one of the focal points of research due to their remarkable performance [2] and relatively simple decoding structure [3]. With great efforts to reduce decoding complexity, increase in hardware speed and the sophistication of very large scale integrated circuits (VLSI) has made the implementation of LDPC codes possible. Moreover, LDPC codes has been adopted in the current IEEE physical layer standards [4], [5], as an optional error correction coding scheme.

Compared to the effort given to reduce the decoder complexity, less attention has been paid to encoding methods. Previously, encoding was a complicated process than decoding due to the random structure of LDPC codes. However, quasi-cyclic low-density parity-check codes (QC-LDPC) [6], [7] as a class of structured LDPC codes were introduced recently to make LDPC codes feasible even for long codewords. The quasi-cyclic LDPC codes has memory efficiency due to their algebraic structure, compared to general LDPC codes. It can be encoded in linear time with shift registers with required memory for storing QC-LDPC codes reduced by a factor 1/Z, if Z X Z circulant permutation matrices are used. The encoding complexity has become almost linear in time by the introduction of the approximate lower triangulation of the parity-check matrix [8], and it has become even more efficient to support faster systematic encoding by having a simple dual diagonal structure without sacrificing its overall performance [9]. The systematic encoding schemes in [8], [9] suggests finding the parity bits directly.

Given the intuitive assumption that encoding operation is based on binary arithmetic, parity bits of LDPC codes can be ultimately found through back-substitutions and bit-inversions. Thus, we propose to bypass finding the initial parity vector to reduce encoding complexity and latency, and find the rest of the parity bits through vector correction technique.

Notation: a boldface letter denotes a vector or a matrix; (·) denotes a vector; (∧) denotes the transpose.

2. Approximate Lower Triangular QC-LDPC Codes with Dual-Diagonal Parity Structure with Single Weight-Three Column

In QC-LDPC codes, the parity-check matrix H can be partitioned into circulant permutation matrix (sub-blocks) of size Z X Z. For example in [4], three sub-block sizes are suggested, as Z = 27, Z = 54, and Z = 81. Let Zmn be Z X Z zero sub-block or identity matrix I with permutation located at the m-th row and the n-th column with a times cyclic shift, 0 ≤ a < Z, to the right. With basic sub-block matrices Z, mZ x nZ parity-check matrix H can be composed of

\[
H = \left[ \begin{array}{ccc}
Z_{0,0}^a & \cdots & Z_{0,n-1}^a \\
\vdots & \ddots & \vdots \\
Z_{m-1,0}^a & \cdots & Z_{m-1,n-1}^a
\end{array} \right] = [H_s|H_p].
\] (1)

As the parity-check matrix is designed for systematic code only, the matrix H again can be divided into two regions. Hs is the sub-matrix for the region where systematic bits are multiplied and Hp represents the parity bit region. The QC-LDPC codes suggested in the latest high throughput PHY standards such as [4], [5] are systematic, i.e., it encodes an information block of size k, \( s = (s_0, s_1, \ldots, s_{k-1})^T \) into a codeword vector \( c = (s_0, s_1, \ldots, s_{k-1}, p_0, p_1, \ldots, p_{n-k-1})^T = (s^T, p_s^T, p_p^T)^T \) of size n, by adding \( n-k \) parity bits obtained so that it must satisfy

\[
Hc = 0.
\] (2)

The vectors \( p_s \) and \( p_p \) are defined as \( (p_0, p_1, \ldots, p_{Z-1})^T \) and \( (p_Z, p_{Z+1}, \ldots, p_{n-k-1})^T \), respectively. The QC-LDPC codes
in [4], [5], [9] have dual diagonal parity structure, and the parity of portion of matrix \( H_p \) can be further decomposed into four sub-matrices as

\[
H_p = \begin{bmatrix} B & T \\ D & E \end{bmatrix}
\]

(3)

where \( I^0 \) denotes identity matrix \( I_{Z \times Z} \) with zero cyclic shift. \( \epsilon \) in \( I^\epsilon \) denotes the value of cyclic shift. The 0 in \( H_p \) means null (zero) sub-matrices which have no edges connecting check to variable nodes. Furthermore, \( V = [B^T \ D^T]^T \) is the weight three columns in the parity generation region.

3. Encoding Procedures for Approximate Lower Triangular QC-LDPC Codes

3.1 Conventional Fast Encoding Scheme by Simple Parity-Check Matrix

The efficient encoding method proposed by Richardson, et al. [8] assumes \( H \) as an approximate lower triangular form. The parity-check matrix \( H \) is in the form

\[
H = [H_i|H_p] = \begin{bmatrix} A & B & T \\ C & D & E \end{bmatrix},
\]

(4)

where the size of \( A \) is \((m-1)\times(n-m)\), \( Z \times (m-1) \), \( T \) is \((m-1)\times(m-1)\), \( C \) is \( Z \times (n-m) \), \( D \) is \( Z \times Z \), and, finally, \( E \) is \( Z \times Z(m-1) \) for QC-LDPC codes defined in [4]. Thus, sub-matrices \( A \) and \( C \) correspond to systematic part \( H_s \), and sub-matrices \( B, D, T, E \) belong to parity bit generation part \( H_p \). Note that \( T \) is lower triangular with identity matrices along the diagonal. By some manipulations of matrix operations, the sectioned \( H \) matrix in (4) related to codeword vector \( c \) in (2), \( c = (s^T, p_{a1}^T, p_{b1}^T)^T \), is summarized as [8]

\[
As + Bp_a + Tp_b = 0
\]

(5)

\[
ET^{-1}As + Cs + ET^{-1}Bp_a + Dp_b = 0,
\]

(6)

where \( ET^{-1} \) is a \((m-1)\times Z \) sub-matrix composed of identity matrix \( I \), which accumulates rows of sub-matrix \( A \). Note that \( ET^{-1}B + D = I \), since addition of all sub-block matrices at weight three part of matrix \( H_p \) suggested in the IEEE 802.11n standard [4] results simply \( Z \times Z \) identity matrix \( I \). Solving (5) and (6) leads to direct solution of parity vectors \( p_a \) and \( p_b \). Thus, each parity bit vectors can be derived as [8]

\[
p_a = \phi^{-1}ET^{-1}As + \phi^{-1}Cs
\]

(7)

\[
p_b = T^{-1}As + T^{-1}Bp_a.
\]

(8)

Since \( \phi^{-1} = ET^{-1}B + D = I^\epsilon \) (i.e., row accumulation of weight three columns), the 1st parity vector \( p_a \) can be found by simplifying (7) to \( p_a = ET^{-1}As + Cs \). Then, the 2nd parity vector \( p_a \) is obtained by block accumulation operation and back substitution starting with plugging \( p_a \) into (8).

3.2 Proposed Novel Vector Correction Method for Lower Computational Complexity

As illustrated in Fig.1, the sub-matrix \( H_p \) can be re-partitioned into three sub-matrices, as

\[
H_p = [V \ Q \ U].
\]

(9)

The purpose of re-partitioning is not on simpler derivation of parity bits generation but is meant to apply different parity vector correction in sectors during the post-processing stage. The sub-matrix \( V \) corresponds to the weight three columns previously described in Sect. 2. The column boundary between \( Q \) and \( U \) is set at \((n - \frac{Z}{2} - 1)\)-th and \((n - \frac{Z}{2})\)-th column. After locating the boundary between \( Q \) and \( U \), a three-step encoding process is proceeded in the followings.

3.2.1 Calculate Vector Cs

Once the information vector \( s \) is given, find vector \( Cs \) for the future use in finding \( p_a \). We note that prior knowledge of cyclic shifted values defined in a sparse matrix \( C \) allows matrix operation \( Cs \) to become an accumulation operation based on shift registers [10].

3.2.2 Set All-Zero Temporary Solution for Parity Vector \( p_a \) and Find Temporary Parity Vector \( p_a^{mp} \) by Back Substitution

Let \( Z \)-all-zero parity bits \( p_{01}, p_{11}, \ldots, p_{Z-1} \) for vector \( p_a \) be generated as a temporal solution. Then, find the remaining parity vector \( p_a^{mp} \) using back substitution procedure [9], realized as

\[
p_a^{mp} = T^{-1}As + B \cdot \text{\( \overrightarrow{0} \)}
\]

\[
= T^{-1}As.
\]

\[
= \begin{bmatrix} \overrightarrow{p_1}^T, \overrightarrow{p_2}^T, \ldots, \overrightarrow{p_{m-1}}^T \end{bmatrix}^T,
\]

(10)

where sub-vector \( \overrightarrow{p_i} \) is defined as \( \overrightarrow{p_i} = [p_{iZ+0}, p_{iZ+1}, \ldots, p_{iZ+\epsilon}] \).
3.2.3 Apply Post Processing to Correct Parity Vector $p^\text{imp}_b$

The assumption of the temporal all-zero vector $p_a$ should be validated. If it is correct, it must be true that right-hand side of the following (12) should sum up to a zero vector, as

$$f = Cs + Ep^\text{imp}_b.$$  

(12)

However, it is very unlikely that the $p^\text{imp}_b$ derived by (10) is correct (i.e. vector $f$ might be a non-zero vector in most cases).

The proposed encoding method suggests the correct solution for $p_a$ is simply $p_a = f$. On the other hand, correct solution for vector $p_b$ is found by adding either $f_\epsilon$ or $f_i$ to $p^\text{imp}_b$ in every sub-block unit $Z$. Here, $f_\epsilon$ and $f_i$ are defined as $I^Tf$ and $(I^T + I^0)f$, respectively. In summary, we obtain the final solution for $p_a$ and $p_b$ as

$$p_a = f$$

$$p_b^O = p^\text{imp}_b + (f_\epsilon, \ldots, f_m)^T, \quad i = 1, 2, \ldots, \frac{m}{2} - 1$$

$$p_b^V = p^\text{imp}_b + (f_\epsilon, \ldots, f_m)^T, \quad i = \frac{m}{2}, \ldots, m - 1$$

$$p_b = [(p_b^O)^T, (p_b^V)^T].$$  

(13)

Note that “$\epsilon$” is the cyclic shift value identical to the one designated in (3).

**Theorem 1:** For a QC-LDPC codes of parity-check matrix $H$ having an almost dual diagonal lower triangular matrix with a weight three column composed of cyclically shifted sub-blocks in the parity generation part defined in (3), the parity bit vector $p_a$ can be obtained by finding $f$, defined in steps from (10) to (11).  

**Proof:** Since $Hc = 0$ should be satisfied, $x + H_p [(p_a)^T, (p_b)^T]^T = 0$ must also be true, where $x$ is defined as

$$x = H_s = [\vec{x}_0, \vec{x}_1, \ldots, \vec{x}_{m-1}]^T.$$  

(14)

The relation of correcting vector $f$ with the direct solution (7) can be elaborated by solving the linear equations $x + H_p [(p_a)^T, (p_b^\text{imp})^T]^T = 0$, expressed as

$$\vec{x}_0 + \vec{p}_0 = \vec{0}$$

$$\vec{x}_1 + \vec{p}_1 + \vec{p}_2 = \vec{0}$$

$$\quad \vdots$$

$$\vec{x}_{m-2} + \vec{p}_{m-2} + \vec{p}_{m-1} = \vec{0}$$

$$\vec{x}_{m-1} + \vec{p}_{m-1} = \vec{0}$$

(15)

Adding all the above binary linear equations would lead to direct solution that was found in [9], and the result is summarized such that

$$\vec{x}_0 + \vec{x}_1 + \ldots + \vec{x}_{m-1} = f$$

$$ET^{-1}As + Cs = f$$

(16)

$$\vec{p}_a = f.$$  

(17)

In contrast to the direct solution for $p_a$ shown in (7), we can acquire it equivalently by finding the correcting vector $f$ which requires operations (10) and (11) only. Note that binary operations $T^{-1}As$ in (10) and Cs in (11) are common to both [9] and the proposed encoding method. In other words, we can avoid unnecessary numerical computations involving the left-hand side of (16) proposed by [9].

**Corollary 1:** Once the $p_a$ is found, correct solution for vector $p^\text{imp}_b$ is obtained by adding either $f_\epsilon$ or $f_i$ in every sub-block unit $Z$, where the former is defined as $f_\epsilon = I^Tf$ and the latter is defined as $f_i = (I^T + I^0)f$.

**Proof:** Given that the true solution for $p_a$ is $f$, vector $p^\text{imp}_b$ is easily found by adding $f_\epsilon$ on both sides of the equation (15) belonging to region “$Q$” in Fig. 1. Therefore, we have

$$\vec{x}_i + I^Tf = \vec{x}_i + I^0f + \vec{p}_{i+1} = 0$$

$$\vec{p}_{i+1} = \vec{x}_i - f_i.$$  

(18)

For the correction of parity vectors belonging to region “$U$” in Fig. 1, it is clear that they are also linearly dependent on $p_a = f$ as well as $I^Tf = f$. Thus, adding an additional vector $f_i$ on the both sides to the $\vec{x}$ + $i$-th row of the corresponding linear equation in (15), we obtain

$$\vec{x}_{m+i} + \vec{x}_a + \vec{p}_{z+i} + \vec{p}_{z+i+1} + f_i = f_i + f$$

$$\vec{x}_{m+i} + \vec{p}_{z+i} + \vec{p}_{z+i+1} + f_i = f_i$$

$$\vec{p}_{z+i+1} + f_i = f_i - \vec{x}_{z+i} - \vec{p}_{z+i}.$$  

(19)

4. Complexity Comparison

In this section, we compare the encoding complexity of our proposed scheme with the Richardson’s scheme [8], [9]. We analyze the number of binary multiplications (AND) and modulo 2 additions (XOR) required during the entire encoding processes. Since generalizing number of ANDs and XORs depends on average number of edges, we compare their complexities through direct numerical analysis respect to the parity-check matrix $H$ having codeword length 1944, $Z=81$, $R=1/2$, details found in [4]. As shown in Table 1, there are operations where both conventional and proposed encoding methods are applied in common. Therefore, we compare computations that are only mutually exclusive to each other (disregarding computations common for both).
In addition, we exclude the computations for \( T^{-1}A \) and \( T^{-1}B \) which can be already found before information vector \( s \) is multiplied. This exclusion is reasonable because pre-computation also applies to the Richardson’s scheme as well which can be assumed that they are stored in a read-only-memory (ROM) format.

Although both methods need to find \( T^{-1}As \) and \( Cs \), the conventional method requires extra row vector accumulation of \( ET^{-1}As \) to find \( f_o \) that is computationally proportional to the average density of rows in sub-matrix \( A \). In contrast, such operation is not needed for the proposed method, but a calculation of \( f \) is required instead. The XOR operation involved with the \( f \) is comparatively less than finding that of \( p_o \). There are 55 non-zero \( Z \times Z \) sub-blocks for matrix \( A \) and 6 non-zero \( Z \times Z \) sub-blocks in \( C \). Thus, the number of AND additions required for finding \( p_o \) by the conventional method is roughly 9 times than that of finding the vector \( f \) by the proposed method.

The extra post-processing computations for the proposed method is trivial. \( f_e \) can be found by cyclically shifting it to the right by \( "e" \). Finding the \( f_e \) requires \( Z \) XOR additions. Moreover, all of the post processing vector correction procedure can be achieved by bit-flipping where elements of \( f, f_e \) and \( f_c \) are “1”. Thus, the total XOR operations, as listed in Table 2, needed for finding \( p_o \) can be further reduced to \( (318 + 1) \times Z \). As summarized in Tables 1 and 2, it is clear that the proposed simple vector correction encoding method requires slightly less complexity than that of the conventional method.

Meanwhile, the overall latency of the proposed method is much lower than that of the conventional method. The conventional method cannot proceed to compute \( p_o \) without the acquisition of \( p_o \), whereas the proposed method can directly go on extracting \( p_o^{mp} \). In addition, it is clear that the time it takes to obtain \( p_o \) by the conventional method requires a longer period than to find \( f \) and add correcting vectors \( f_e \) and \( f_c \) to vector \( p_o^{mp} \).

5. Conclusions

In this letter, we proposed an alternative encoding method for QC-LDPC codes with dual diagonal parity structure with one weight three columns in the parity generation sub-matrix. We have demonstrated that direct approach to determining \( p_o \), lead by the fast encoding scheme [9], is unnecessary. Moreover, the vector correction encoding method is directly applicable to current IEEE 802.11n and IEEE 802.16e standards, while the overall complexity and encoding latency are lower than those of the conventional method.

Acknowledgements

This research was supported by the KCC (Korea Communications Commission), Korea, Under the R&D program supervised by the KCA (Korea Communications Agency) (KCA-2011-09913-04006). The work of Hoojin Lee was financially supported by Hansung University.

References