Decoding of Quasi-Cyclic LDPC Codes With Section-Wise Cyclic Structure

Juane Li, Keke Liu, Shu Lin and Khaled Abdel-Ghaffar
Department of Electrical and Computer Engineering, University of California, Davis, CA 95616, USA
Email: {jueli, kkeliu, shulin, ghaffar}@ucdavis.edu

***Abstract*—Presented in this paper is a reduced-complexity iterative decoding scheme for quasi-cyclic (QC) LDPC codes. This decoding scheme is devised based on the *section-wise cyclic structure* of the parity-check matrix of a QC-LDPC code. Using this decoding scheme, the hardware implementation complexity of a QC-LDPC decoder can be significantly reduced without performance degradation. A high-rate QC-LDPC code that can achieve a very low error-rate without a visible error-floor is used to demonstrate the effectiveness of the proposed decoding scheme. Also presented in this paper are two other high-rate QC-LDPC codes and a method for constructing rate-$\frac{1}{2}$ QC-LDPC codes whose Tanner graphs have girth 8. All the codes constructed perform well with low error-floor using the proposed decoding scheme.**

I. Introduction

LDPC codes [1] with *quasi-cyclic* structure [2], called quasi-cyclic (QC) LDPC codes, are the most preferred type of LDPC codes for practical applications in communication and storage systems. A QC-LDPC code is given by the null space of an *array* $\mathbf{H}$ of *sparse circulant matrices* of the same size over a finite field, binary or nonbinary. In most of the proposed constructions of QC-LDPC codes, the sparse circulant matrices in the parity-check array $\mathbf{H}$ of a QC-LDPC code are *circulant permutation matrices* (CPMs) or *zero matrices* (ZMs). Such a parity-check array $\mathbf{H}$ of a QC-LDPC code is said to possess *CPM-structure*. We also say that the QC-LDPC code has CPM-structure.

QC-LDPC codes with CPM-structure have advantages over other types of LDPC codes in hardware implementation of encoding and decoding. Encoding of a QC-LDPC code with CPM-structure can be efficiently implemented using simple shift registers with complexity linearly proportional to its number of parity-check symbols (or its length) [2], [3]. In hardware implementation of a QC-LDPC decoder, the CPM-structure of the parity-check array of the code simplifies the wire routing for messages passing between the variable-node (VN) message processing units and check-node (CN) message processing units [4].

In this paper, we present an iterative scheme for decoding QC-LDPC codes whose parity-check arrays possess CPM-structure. This decoding scheme significantly reduces the hardware implementation complexity of a QC-LDPC decoder in terms of the number of the CN message processing units and the number of wires required to connect the VN message processing units and the CN message processing units. This decoding scheme can be incorporated with any known reliability updating algorithm, such as the sum-product algorithm (SPA) or the min-sum algorithm (MSA), to form a reduced-complexity iterative decoding algorithm for QC-LDPC codes. A proper combination of the proposed decoding scheme and a specific reliability updating algorithm can result in no performance degradation. Well designed QC-LDPC codes with CPM-structure can perform just as well as any other type of LDPC codes. All these advantages inevitably will make QC-LDPC codes with CPM-structure the mainstream LDPC codes for future applications in communication and storage systems.

The rest of the paper is organized as follows. In Section II, we present the reduced-complexity iterative decoding scheme for QC-LDPC codes with CPM-structure. In Section III, we incorporate the MSA into the proposed decoding scheme to form a reduced-complexity iterative decoding algorithm for QC-LDPC codes with CPM-structure. In Section IV, we present a high-rate high-performance QC-LDPC code with CPM-structure to demonstrate the effectiveness of the proposed decoding scheme with the MSA for updating the reliabilities of the received symbols. We will show that the code can achieve a very low error-rate without a visible error-floor. In the same section, two other high-rate QC-LDPC codes are constructed by shortening the high-rate, high-performance and low error-floor code. Section V presents a method for constructing well performing rate-$\frac{1}{2}$ $(3, 6)$-regular QC-LDPC codes using a masking technique. A $(1040, 520)$ QC-LDPC code whose Tanner graph has girth 8 and contains a small number of cycles of length 8 is constructed to demonstrate the effectiveness of the proposed construction method. Section VI concludes the paper with some remarks.

II. An Iterative Decoding Scheme For QC-LDPC Codes with CPM-Structure

A. A Brief Review of an Earlier Reduced-Complexity Iterative Decoding Scheme for QC-LDPC Codes

Recently, we devised a reduced-complexity iterative decoding scheme for QC-LDPC codes, called a *revolving iterative decodig (RID) scheme* [5]. This decoding scheme significantly reduces the hardware complexity of a QC-LDPC decoder in terms of the number of the CN message processing units and the number of wires required to connect the VN message processing units and the CN message processing units. To apply this decoding scheme for a QC-LDPC code, its parity-check array $\mathbf{H}$ must have the *row-block cyclic structure*, i.e., each row-block (of submatrices) of $\mathbf{H}$ is the cyclic-shift of the row-block above it to *the right* by a fixed number of positions, say l positions, and the first row-block is the cyclic-shift of the last row-block to the right by l positions. Suppose the parity-check array $\mathbf{H}$ consists of m row-blocks. Then, the entire parity-check array can be generated by cyclically shifting its first row-block $m-1$ times. As a result, iterative decoding of the QC-LDPC code given by the null space of $\mathbf{H}$ can be carried out based on the first row-block of $\mathbf{H}$ rather than the entire $\mathbf{H}$. The RID scheme given in [5], [6] is devised based on this row-block cyclic structure.

However, most of the proposed constructions of QC-LDPC codes are specified by parity-check arrays with CPM-structure, i.e., in the form of arrays of CPMs and/or ZMs. These arrays, in general, do not have the row-block cyclic structure. To acquire the row-block cyclic structure, specific column and row permutations must be performed to transform an array $\mathbf{H}$ of CPMs and/or ZMs into another array $\tilde{\mathbf{H}}$ of submatrices of the same size [5], [6]. The permuted array $\tilde{\mathbf{H}}$ possesses row-block cyclic structure but its nonzero submatrices are no longer CPMs. Consequently, in the hardware implementation of an RID-decoder based on the row-block cyclic structure of $\tilde{\mathbf{H}}$, the advantages of simple wire routing due to the CPM-structure of the parity-check array $\mathbf{H}$ are lost.

B. A Revolving Iterative Decoding Scheme Based on the Section-Wise Cyclic Structure

In the following, we present a variation of the RID scheme, called the CPM-RID scheme, for the QC-LDPC codes whose parity-check arrays possess CPM-structure, i.e., they are arrays of CPMs and/or ZMs. The CPM-RID scheme is simply devised based on the CPM-structure of the parity-check array $\mathbf{H}$ of a QC-LDPC code *without column and row permutations*. This CPM-RID scheme not only has the same advantage in reduction of the decoding hardware complexity as the RID scheme proposed in [5], [6] but also maintains the advantage of simple wire routing due to the CPM-structure [4]. In our presentation of the CPM-RID scheme, we only consider the decoding of binary QC-LDPC codes. The scheme can be generalized for decoding a q-ary QC-LDPC code over GF(q) whose parity-check array consists of α-multiplied CPMs [2], [7], [8] where α is a primitive element of GF(q) and q is a prime or a power of a prime.

Let $\mathbf{H}$ be an $m \times n$ array of CPMs and/or ZMs of size $(q-1) \times (q-1)$. Let $\mathbf{H}_0$, $\mathbf{H}_1$, …, $\mathbf{H}_{m-1}$ denote the m row-blocks of $\mathbf{H}$, each consisting of n CPMs and/or ZMs of size $(q-1) \times (q-1)$. For $0 \leq i < m$ and $0 \leq k < q-1$, let $\mathbf{h}_{i,k} = (\mathbf{h}_{i,k,0}, \mathbf{h}_{i,k,1}, \ldots, \mathbf{h}_{i,k,n-1})$ be the k-th row in the i-th row-block $\mathbf{H}_i$ which consists of n sections, $\mathbf{h}_{i,k,0}, \mathbf{h}_{i,k,1}, \ldots, \mathbf{h}_{i,k,n-1}$, each containing $q-1$ components. Each section is the k-th row of either a CPM or a ZM in $\mathbf{H}_i$. If we cyclically shift all the n sections of $\mathbf{h}_{i,k}$ simultaneously one place to the right *within the sections*, we obtain the $(k+1)$-th row $\mathbf{h}_{i,k+1}$ of $\mathbf{H}_i$. For $0 \leq j < n$, the j-th section of $\mathbf{h}_{i,k+1}$ is the $(k+1)$-th row of either a CPM or a ZM in the i-th row block $\mathbf{H}_i$ of $\mathbf{H}$. The above cyclic-shift within each section is referred to as *section-wise cyclically-shifting* of the row $\mathbf{h}_{i,k}$. For $k = q-2$, the section-wise cyclic-shift of $\mathbf{h}_{i,q-2}$ results in the 0-th row $\mathbf{h}_{i,0}$ of $\mathbf{H}_i$. Consequently, all the rows of the i-th row-block $\mathbf{H}_i$ can be obtained by section-wise cyclically shifting the 0-th row $\mathbf{h}_{i,0}$ $q-2$ times. This section-wise cyclically shifting of the rows of $\mathbf{H}$ maintains the CPM-structure of each row-block of $\mathbf{H}$.

Let $\mathbf{H}_0^\star$ be an $m \times n(q-1)$ matrix which consists of the first (or the top) rows $\mathbf{h}_{0,0}$, $\mathbf{h}_{1,0}$, …, $\mathbf{h}_{m-1,0}$ of the m row-blocks $\mathbf{H}_0$, $\mathbf{H}_1$, …, $\mathbf{H}_{m-1}$ of the parity-check array $\mathbf{H}$. Then, it follows from the section-wise cyclic structure of $\mathbf{H}$, the entire array $\mathbf{H}$ can be obtained by section-wise cyclically shifting $\mathbf{H}_0^\star$ $q-2$ times. This section-wise cyclic structure allows us to decode the QC-LDPC code given by the parity-check array $\mathbf{H}$ based on the submatrix $\mathbf{H}_0^\star$ *alone* in almost the same way as the RID scheme given in [5], [6].

Each decoding iteration based on $\mathbf{H}_0^\star$ is called a *decoding sub-iteration*. At the end of each decoding sub-iteration, the reliabilities of the received symbols are updated with a chosen reliability updating algorithm. Then, the reliability vector ($n(q-1)$ components in n sections) and the received sequence ($n(q-1)$ symbols in n sections) are section-wise cyclically shifted to the left by one position and used as the input information (maybe together with the channel information) to carry out the next decoding sub-iteration based on $\mathbf{H}_0^\star$. It is clear that any $q-1$ decoding sub-iterations performed based on $\mathbf{H}_0^\star$ are *equivalent to one decoding iteration based on the entire parity-check array* $\mathbf{H}$. At the end of each decoding sub-iteration, the syndrome $\mathbf{s}$ of the hard-decision of the received sequence (after section-wise cyclically shifted one position to the left) is computed based on the entire parity-check array $\mathbf{H}$. If $\mathbf{s} = \mathbf{0}$, we stop the decoding process; otherwise, we continue the decoding process until a preset maximum number of decoding sub-iterations is reached. A flow chart of the decoding scheme is given in Fig.1.

The above decoding process of a QC-LDPC code simply revolves around $\mathbf{H}_0^\star$ iteratively. Since the CPM-structure of the parity-check array $\mathbf{H}$ is preserved, the wire routing advantage due to the CPM-structure of the parity-check array [4] is maintained. This revolving iterative decoding (RID) scheme is simply a variation of the RID scheme presented in [6] and thus we call it CPM-RID scheme. The submatrix $\mathbf{H}_0^\star$ is called the *decoding matrix*.

Based on the structure of $\mathbf{H}_0^\star$, we can easily see that the number of rows and the number of 1-entries in $\mathbf{H}_0^\star$ are $\frac{1}{q-1}$-th of those of $\mathbf{H}$. Therefore, implementing a CPM-RID decoder of a QC-LDPC code with CPM-structure, the number of CN message processing units and the number of wires required to connect the CN message processing units and VN message processing units are reduced by *a factor of* $q-1$ of those required in implementing a QC-LDPC decoder based on the entire parity-check array $\mathbf{H}$ of the code. For large q, there is a tremendous reduction in hardware implementation complexity of a QC-LDPC decoder using the CPM-RID scheme. Furthermore, the decoder still enjoys the simple wire routing advantage due to the CPM-structure.

III. A Min-Sum CPM-RID Algorithm

Any known reliability updating algorithm can be incorporated in the CPM-RID scheme to decode a QC-LDPC code with CPM-structure. In this section, we use the MSA for updating reliabilities of the received symbols in each decoding sub-iteration of the CPM-RID scheme. The incorporation of the MSA into the CPM-RID scheme can be achieved using exactly the same approach as in [5], [6]. This combination of the CPM-RID and the MSA is called *CPM-RID-MSA (CPM-RMSA)*. In updating the reliabilities of the received symbols, there is a major difference between the CPM-RMSA and the RMSA proposed in [5], [6]. This will be explained later.

Let $\mathbf{y} = \big(y_0, y_1, \ldots, y_{n(q-1)-1}\big)$ be the soft-decision received vector at the output of the receiver detector. We assume that the symbols of a codeword are transmitted over a binary AWGN channel with BPSK signaling. Let I_{max} be the maximum number of iterations to be performed based on the entire parity-check array $\mathbf{H}$, each iteration consisting of $q-1$ decoding sub-iterations based on the $m \times n(q-1)$ submatrix $\mathbf{H}_0^\star = [h_{t,j}]_{0 \le t < m, 0 \le j < n(q-1)}$. Therefore, to perform I_{max} iterations based on $\mathbf{H}$ is equivalent to perform $I_{max}(q-1)$ decoding sub-iterations based on $\mathbf{H}_0^\star$. We label the iterations from 0 to $I_{max}-1$. For each iteration, we label its sub-iterations from 0 to $q-2$. We use the symbols i and k to denote the k-th sub-iteration of the i-th iteration, respectively. For $0 \le j < n(q-1)$, $0 \le t < m$ and $0 \le k < q-1$, let $L_{t\to j}^{(k)}$ denote the message sent from the t-th CN to the j-th VN in the Tanner graph of $\mathbf{H}_0^\star$ in the k-th sub-iteration of each iteration. For $0 \le j < n(q-1)$ and $0 \le t < m$, let $N(j) = \{t : 0 \le t < m, h_{t,j} = 1\}$ and $M(t) = \{j : 0 \le j < n, h_{t,j} = 1\}$. Let $\mathbf{R} = (R_0, R_1, \ldots, R_{n(q-1)-1})$ denote the reliability information vector. Let λ_{atten} denote the attenuation factor for a scaled MS-based reliability information updating.

With the above definitions and notations, the CPM-RMSA is formulated as follows:

CPM-RMSA

Step 1. (**Initialization**) For $0 \le j < n(q-1)$, set $R_j = y_j$. For $0 \le j < n(q-1)$, $t \in N(j)$ and $0 \le k < q-1$, set $L_{t\to j}^{(k)} = 0$. Set $i = 0$.

Step 2. (**Iteration**) Set $k = 0$. Carry out the k-th sub-iteration in the i-th iteration as follows:

(1) Update reliability information $\mathbf{R}$ as follows: for $0 \le j < n(q-1)$,

$$R_j \leftarrow R_j - \sum_{t \in N(j)} L_{t\to j}^{(k)}, \tag{1}$$

and use the updated R_j as the outgoing VN messages.

(2) For $0 \le j < n(q-1)$ and $t \in N(j)$, update the outgoing CN messages $L_{t\to j}^{(k)}$ as follows:

$$L_{t\to j}^{(k)} = \lambda_{atten} \cdot \left[\prod_{j' \in M(t)\setminus j} \text{sign}\left(R_{j'}\right)\right] \times \left[\min\{|R_{j'}| : j' \in M(t) \setminus j\}\right]. \tag{2}$$

(3) Update reliability information $\mathbf{R}$ as follows: for $0 \le j < n$,

$$R_j \leftarrow R_j + \sum_{t \in N(j)} L_{t\to j}^{(k)}. \tag{3}$$

(4) Form the hard decision vector $\mathbf{z}$ based on $\mathbf{R}$; section-wise cyclically shift $\mathbf{z}$ by k positions to the left and compute its syndrome $\mathbf{s} = \mathbf{z} \cdot \mathbf{H}^T$. If $\mathbf{s} = \mathbf{0}$, go to Step 4; Otherwise, go to Step 2-(5).

(5) Section-wise cyclically shift $\mathbf{R}$ one position to the left. If $k < q-2$, set $k \leftarrow k+1$ and return to Step 2-(1) to begin another sub-iteration.

Step 3. If $i = I_{max} - 1$, stop decoding and declare a decoding failure. Otherwise, set $i \leftarrow i+1$ and return to Step 2.

Step 4. Output $\mathbf{z}$ as the decoded codeword.

Here, we point out the major difference between the above CPM-RMSA and the RMSA proposed in [5], [6]. Using the RMSA, each VN sends the same message to all its adjacent CNs in the Tanner graph of $\mathbf{H}_0^\star$ *without subtracting the CN message received at the end of previous iteration*. This may result in noticeable performance degradation compared to the decoding based on the entire parity-check array $\mathbf{H}$ using the conventional MSA if the *row redundancy* (defined as the number of the redundant rows) of $\mathbf{H}$ is small or zero. In the CPM-RMSA, each VN still sends the same message to all its adjacent CNs in the Tanner graph of $\mathbf{H}_0^\star$. However, the message sent from a VN is the difference between the updated reliability message at the VN and the sum of the messages previously received from all its adjacent CNs in the Tanner graph of $\mathbf{H}_0^\star$. Since the previous CN messages are removed from the updated reliability messages, the CPM-RMSA gives a better error performance for decoding QC-LDPC codes whose parity-check arrays have small or no row redundancies. This is especially the case for high-rate QC-LDPC codes which will be demonstrated in the next section.

IV. High-Rate High-Performance QC-LDPC Codes Decoded With the CPM-RMSA

Consider the prime field GF(131) with 131 elements. Let α be a primitive element of this field. Form the following 6×124 matrix over GF(131) based on two disjoint subsets of the nonzero elements of GF(131), $\boldsymbol{S}_1 = \left\{\alpha^0 = 1, \alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^5\right\}$ and $\boldsymbol{S}_2 = \left\{\alpha^6, \alpha^7, \ldots, \alpha^{128}, \alpha^{129}\right\}$:

$$\mathbf{B} = \begin{bmatrix} \alpha^0+\alpha^6 & \alpha^0+\alpha^7 & \cdots & \alpha^0+\alpha^{128} & \alpha^0+\alpha^{129} \\ \alpha^1+\alpha^6 & \alpha^1+\alpha^7 & \cdots & \alpha^1+\alpha^{128} & \alpha^1+\alpha^{129} \\ \alpha^2+\alpha^6 & \alpha^2+\alpha^7 & \cdots & \alpha^2+\alpha^{128} & \alpha^2+\alpha^{129} \\ \alpha^3+\alpha^6 & \alpha^3+\alpha^7 & \cdots & \alpha^3+\alpha^{128} & \alpha^3+\alpha^{129} \\ \alpha^4+\alpha^6 & \alpha^4+\alpha^7 & \cdots & \alpha^4+\alpha^{128} & \alpha^4+\alpha^{129} \\ \alpha^5+\alpha^6 & \alpha^5+\alpha^7 & \cdots & \alpha^5+\alpha^{128} & \alpha^5+\alpha^{129} \end{bmatrix}. \tag{4}$$

It can be easily proved that the above matrix $\mathbf{B}$ has the following structural properties: (1) all the entries in a row (or in a column) of $\mathbf{B}$ are distinct elements in GF(131); (2) any two rows (or two columns) in $\mathbf{B}$ differ at every position; (3) all the entries are nonzero elements in GF(131); and (4) any 2×2 submatrix of $\mathbf{B}$ is *nonsingular*.

Express $\mathbf{B}$ in the form of $\mathbf{B} = [b_{i,j}]_{0 \leq i < 6, 0 \leq j < 124}$. Each entry $b_{i,j}$ in $\mathbf{B}$ is a power of α, say $b_{i,j} = \alpha^{l_{i,j}}$, where $0 \leq l_{i,j} < 130$. Let $\mathbf{A}$ be a 130×130 CPM over GF(2). We label both rows and columns of $\mathbf{A}$ from 0 to 129. Each row (or each column) of $\mathbf{A}$ consists of a single 1-component. Each row is the *cyclic-shift (one place to right)* of the row above it and the first row is the cyclic-shift of the last row one place to the right. It is clear that a CPM is uniquely specified by the location of the unique 1-component of its top row (or the 0-th row). There are exactly 130 distinct CPMs of size 130×130. Now, we represent the entry $b_{i,j} = \alpha^{l_{i,j}}$ in $\mathbf{B}$ by the 130×130 CPM, denoted by $\mathbf{A}(\alpha^{l_{i,j}})$, in which the single 1-component of its top row is at the location $l_{i,j}$. This matrix representation is referred to as the *CPM-dispersion* of the field element $\alpha^{l_{i,j}}$ [2], [8], [9]. Since there are 130 nonzero elements in GF(131) and there are exactly 130 different CPMs over GF(2) of size 130×130, there is a *one-to-one correspondence* between a nonzero element of GF(131) and a CPM of size 130×130. Therefore, each nonzero element of GF(131) is uniquely represented by a CPM of size 130×130.

The above replacement of each entry in the 6×124 matrix $\mathbf{B}$ by a 130×130 CPM results in a 6×124 array $\mathbf{H}$ of CPMs of size 130×130. The array $\mathbf{H}$ consists of 6 row-blocks and 124 column-blocks of CPMs. Each row-block consists of 124 CPMs of size 130×130 and each column-block consists of 6 CPMs of size 130×130. The array $\mathbf{H}$ is a 780×16120 matrix over GF(2) with constant column weight 6 and row weight 124, and is called the *CPM-dispersion* of $\mathbf{B}$. The matrix $\mathbf{B}$ is called the *base matrix* for the dispersion [2], [8], [9].

The rank of $\mathbf{H}$ is 775 and there are 5 redundant (or dependent) rows in $\mathbf{H}$. The null space of $\mathbf{H}$ gives a $(6, 124)$-regular binary $(16120, 15345)$ QC-LDPC code C of length 16120 with rate 0.9519 (a very high-rate code). Let $\mathcal{G}$ be the Tanner graph of C. Since every 2×2 submatrix of $\mathbf{H}$ is nonsingular, it follows from Proposition 1 given in [9] that the girth of the Tanner graph $\mathcal{G}$ of C is at least 6.

Using the algorithm given in [10], we find that the Tanner graph $\mathcal{G}$ of C has girth exactly 6 and it contains $37,796,720$ cycles of length 6, a very large number of short cycles. Each VN of the Tanner graph $\mathcal{G}$ is on 7034 cycles of length 6. On a cycle of length 6, each VN is connected to two other VNs by paths of length 2. Two cycles of length 6 may have two common VNs. Therefore, for a VN-v, there are at least $7,034$ other VNs connected to it by paths of length 2, a very high-degree of *connectivity* [11]. This high-degree of connectivity allows *rapid exchange of information* between all the VNs. Consequently, in a few decoding iterations, each VN processing unit collects enough information from the other VN processing units to update the reliability of the VN being processed to a level to make a correct hard-decision with high probability. As a result, the decoding process can converge into a codeword in a *small number* of decoding iterations. The large number of cycles of length 6 contained in the Tanner graph $\mathcal{G}$ of the constructed code, of course, causes correlation in the message passing during iterative decoding (after a few iterations). However, the rapid exchange of information between all VNs due to the high-connectivity between all the VNs of the Tanner graph offsets the effect of the large number of cycles of length 6.

First, we decode this code with a scaled MSA [2], [12] based on the entire parity-check matrix $\mathbf{H}$. To implement a decoder based on $\mathbf{H}$, it requires 780 CN message processing units and $96,720$ wires to connect the 780 CN message processing units and $16,120$ VN processing units (a very large number of wires). The bit and block error performances of the code decoded with 5, 10 and 50 iterations of the MSA with a scaling factor 0.75 (optimized) are shown in Fig.2 (The performances are computed by an FPGA decoder.). From Fig.2, we see that, with 50 iterations, the code achieves a BER (bit-error rate) of 10^{-13} without a visible error-floor. It also has a beautiful waterfall performance. At the BER of 10^{-13}, the code performs 1.33 dB away from the Shannon limit. It achieves a 8.71 dB coding gain over the uncoded BPSK system. Next, we see that the decoding of the code converges very fast. At the BER of 10^{-10}, the performance gap between 5 and 10 decoding iterations is about 0.2 dB and performance gap between 10 and 50 decoding iterations is less than 0.1 dB.

Next, we decoded the code with the CPM-RMSA with a scaling factor 0.5 (optimized). The submatrix $\mathbf{H}_0^\star$ used to carry out the decoding consists of 6 rows, each being the first row of a row-block of $\mathbf{H}$. It is a 6×16120 submatrix of $\mathbf{H}$. The number of 1-entries in $\mathbf{H}_0^\star$ is 744. To implement a CPM-RMSA decoder based on $\mathbf{H}_0^\star$, it requires only 6 CN message processing units and 744 wires to connect the 6 CN message processing units to the 744 VN message processing units.

We see that, using the CPM-RMSA based on $\mathbf{H}_0^\star$ to decode the $(16120, 15345)$ QC-LDPC code, there is a factor of 130 reduction in decoding complexity compared to using the conventional MSA based on the entire parity-check matrix $\mathbf{H}$ to decode the code. The reduction in decoder hardware complexity is huge.

The error performances of the code decoded with 5, 10 and 50 iterations of the CPM-RMSA (equivalent to 650, 1300 and 6500 sub-iterations based on $\mathbf{H}_0^\star$) are shown in Fig.3. We see that the decoding of the code with the CPM-RMSA also converges very fast. A comparison of the error performances of the code decoded with 50 iterations of the CPM-RMSA and decoded with 50 iterations of the scaled MSA is shown in Fig.4. We see that the performance curves of the code decoded with the CPM-RMSA and the scaled MSA, respectively, basically overlap with each other.

From the above example, we see that the CPM-RMSA not only reduces decoder complexity but also performs as well as the conventional MSA. This large reduction in complexity also results in a larger reduction in power consumption. Complexity and power consumption are two critical issues in practical applications in many communications and storage systems.

QC-LDPC codes of *various lengths and rates* can be constructed by using submatrices of the matrix $\mathbf{B}$ given by (4) as base matrices. Suppose we delete the last 34 columns from $\mathbf{B}$. This deletion results in a 6×90 submatrix $\mathbf{B}(6,90)$ of $\mathbf{B}$. The CPM-dispersion of $\mathbf{B}(6,90)$ gives a 6×90 array $\mathbf{H}(6,90)$ of CPMs of size 130×130 which is a 780×11700 matrix over GF(2) with column and row weights, 6 and 90, respectively. The null space of $\mathbf{H}(6,90)$ gives a $(11700, 10925)$ QC-LDPC code C_1 with rate 0.9337, another high-rate code. The bit and block error performances of C_1 decoded with 5, 10 and 50 iterations of the CPM-RMSA are shown in Fig.5. We see that the decoding of this code with the CPM-RMSA converges very fast.

Suppose we delete the last 60 columns and the last 2 rows from $\mathbf{B}$. This results in a 4×64 submatrix $\mathbf{B}(4,64)$ of $\mathbf{B}$. The CPM-dispersion of $\mathbf{B}(4,64)$ gives a 4×64 array $\mathbf{H}(4,64)$ of CPMs of size 130×130 which is a 520×8320 matrix with column and row weights, 4 and 64, respectively. The null space of the $\mathbf{H}(4,64)$ gives a $(8320, 7803)$ QC-LDPC code C_2 with rate 0.9378. The bit and block error performances of C_2 decoded with 5, 10 and 50 iterations of the CPM-RMSA are shown in Fig.6. We see that the decoding of this code with the CPM-RMSA also converges very fast.

The CPM-RMSA decoder for the $(16120, 15345)$ QC-LDPC code C can be used to decode the two shortened codes C_1 and C_2 of C by simply de-activating the VN message processing units corresponding to the columns in the column-blocks and/or the CN message processing units corresponding to the rows in the row-blocks of the 6×124 parity-check array $\mathbf{H}$ of C which are being deleted.

V. Rate-$\frac{1}{2}$ Codes

QC-LDPC codes constructed in the last section using the matrix $\mathbf{B}$ given by (4) and its submatrices as base matrices are long and high-rate codes. However, low-rate QC-LDPC codes of short lengths with good error performances can also be constructed by taking small submatrices of $\mathbf{B}$ as base matrices and *masking* [2], [8], [13] them.

Masking a submatrix $\mathbf{B}^\star$ of $\mathbf{B}$ is done by replacing *nonzero entries* at certain chosen locations in $\mathbf{B}^\star$ with zeros. In the CPM-dispersion $\mathbf{H}^\star$ of $\mathbf{B}^\star$, this replacement results in replacing CPMs in $\mathbf{H}^\star$ by zero matrices (ZMs) at the locations corresponding to the locations in $\mathbf{B}^\star$ where nonzero entries are replaced by zeros. Since the base matrix $\mathbf{B}$ given by (4) contains only nonzero entries, any submatrix $\mathbf{B}^\star$ of $\mathbf{B}$ also contains only nonzero entries. Let λ be a nonnegative integer less than the number of total entries in $\mathbf{B}^\star$. The replacement of λ nonzero entries in $\mathbf{B}^\star$ by λ zeros amounts to replacing λ CPMs by λ ZMs at the locations in $\mathbf{H}^\star$ corresponding to the locations of the λ nonzero entries in $\mathbf{B}^\star$ which are being replaced by zeros. Masking λ CPMs in $\mathbf{H}^\star$ amounts to removing 130λ edges from the Tanner graph $\mathcal{G}^\star$ associated with $\mathbf{H}^\star$. Removing these edges in $\mathcal{G}^\star$ may break many short cycles in $\mathcal{G}^\star$. As a result, the resultant Tanner graph $\mathcal{G}^\star_{mask}$ may have a much smaller number of short cycles, or a larger girth, or both. The subscript *"mask"* stands for masking. In choosing the nonzero entries in $\mathbf{B}^\star$ to be masked, we have to avoid disconnecting the Tanner graph of $\mathbf{H}^\star$ (or making some columns or rows of $\mathbf{H}^\star_{mask}$with very small weights).

For $1 \leq k \leq 6$ and $1 \leq l \leq 124$, let $\mathbf{B}(k,l) = [b_{i,j}]_{0 \leq i<k, 0 \leq j<l}$ be a $k \times l$ submatrix of $\mathbf{B}$. The operation of masking $\mathbf{B}(k,l)$ can be modeled mathematically. Let $\mathbf{Z}(k,l) = [z_{i,j}]_{0 \leq i<k, 0 \leq j<l}$ be a $k \times l$ matrix with the zero element and unit element of GF(131) as entries. Define the following product of $\mathbf{Z}(k,l)$ and $\mathbf{B}(k,l)$: $\mathbf{B}_{mask}(k,l) = \mathbf{Z}(k,l) \otimes \mathbf{B}(k,l) = [z_{i,j} b_{i,j}]_{0 \leq i<k, 0 \leq j<l}$ where $z_{i,j} b_{i,j} = b_{i,j}$ if $z_{i,j}=1$ and $z_{i,j} b_{i,j} = 0$ if $z_{i,j}=0$. In this matrix product (known as *Hadamard product* [14]) operation, nonzero entries in $\mathbf{B}(k,l)$ at the locations corresponding to the locations of zero-entries in $\mathbf{Z}(k,l)$ are replaced (or masked) by 0's. The CPM-dispersion of $\mathbf{B}_{mask}(k,l)$ gives a $k \times l$ masked array $\mathbf{H}_{mask}(k,l)$ of CPMs and ZMs of size 130×130. We call $\mathbf{Z}(k,l)$ and $\mathbf{B}_{mask}(k,l)$ the *masking matrix* and the *masked matrix*, respectively.

Suppose we take a 4×8 submatrix $\mathbf{B}(4,8)$ from $\mathbf{B}$ and mask it with the following 4×8 masking matrix:

$$\mathbf{Z}(4,8) = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{5}$$

The masking results in a 4×8 masked matrix $\mathbf{B}_{mask}(4,8)$ with constant column weight 3 and constant row weight 6. The CPM-dispersion of $\mathbf{B}_{mask}(4,8)$ is a 4×8 masked array $\mathbf{H}_{mask}(4,8)$ of CPMs and ZMs of size 130×130, each column-block consisting of 3 CPMs and 1 ZM and each row-block consisting of 6 CPMs and 2 ZMs. The masked array $\mathbf{H}_{mask}(4,8)$ is a 520×1040 matrix with column and row weight, 3 and 6, respectively. The null space of $\mathbf{H}_{mask}(4,8)$ gives a $(3,6)$-regular $(1040, 520)$ QC-LDPC code of rate-$\frac{1}{2}$ if $\mathbf{H}_{mask}(4,8)$ is a full rank matrix.

As an example, we consider the submatrix $\mathbf{B}(4,8)$ of $\mathbf{B}$ with entries confined in the first four rows of $\mathbf{B}$ and 8 consecutive columns of $\mathbf{B}$ from the 38-th column to the 45-th column. Masking this submatrix $\mathbf{B}(4,8)$ with the matrix $\mathbf{Z}(4,8)$ given by (5), we obtain a 4×8 masked matrix $\mathbf{B}_{mask}(4,8)$. Using this masked matrix $\mathbf{B}_{mask}(4,8)$ as the base matrix, we obtain a $(3,6)$-regular $(1040, 520)$ QC-LDPC code C_3 of rate-$\frac{1}{2}$. Using the cycle enumeration algorithm given in [10], we find that the Tanner graph $\mathcal{G}$ of C_3 has girth 8 and contains 390 cycles of length 8, $13,260$ cycles of length 10. Each VN of the Tanner graph $\mathcal{G}$ is on 2 cycles of length 8. On a cycle of length 8, each VN is connected to two other VNs by paths of length 2. Therefore, for a VN-v, there are 4 other VNs connected to it by paths of length 2. The number of 1-entries in $\mathbf{H}^\star_0$ corresponding to C_3 is 24. To implement a CPM-RMSA decoder based on $\mathbf{H}^\star_0$, it requires only 4 CN message processing units and 24 wires to connect the 4 CN message processing units to the 24 VN message processing units. The bit and block error performance of this code decoded with CPM-RMSA are shown in Fig.7. Also included in Fig.7 are the bit and block error performances of a code C_{peg} constructed by the PEG algorithm [2], [15]. Using the cycle enumeration algorithm given in [10], we find that the Tanner graph of C_{peg} has girth 8 and contains 6 cycles of length 8, $11,958$ cycles of length 10. C_{peg} is not in quasi-cyclic form, and it has constant column weight 3, three row weights 5, 6 and 7. From Fig.7, we see that the (1040,520) QC-LDPC code C_3 decoded with the CPM-RMSA performs very well compared to the PEG code C_{peg}. The two bit error performance curves of the code decoded with 10 and 50 iterations of the CPM-RMSA basically overlap with each other. And the code with 10 iterations of CPM-RMSA achieves the BER of 10^{-8} without a visible error-floor. Since the PEG code C_{peg} is not quasi-cyclic, it can not be decoded with the CPM-RMSA. It is decoded with the conventional MSA with scaling factor 0.75. Therefore, the decoding complexity of C_3 is much less than that of the PEG code C_{peg}.

There are 55 other 4×8 submatrices of $\mathbf{B}$. Masking these submatrices with the masking matrix given by (5) results in masked matrices whose CPM-dispersions give rate-$\frac{1}{2}$ QC-LDPC codes with girth 8. They all perform well with the CPM-RMSA. The masking matrix given by (5) is an effective masking matrix to produce Tanner graphs with girth 8.

QC-LDPC codes of other rates of short lengths can also be constructed by using submatrices of $\mathbf{B}$ given by (4).

VI. Conclusion and Remarks

In this paper, we presented a reduced-complexity iterative decoding scheme for QC-LDPC codes with CPM-structure. This decoding scheme not only reduces hardware implementation complexity of the decoder, but also performs well. It may find applications in many communication and storage systems where low complexity and low power consumption are critical. The proposed decoding scheme can be generalized for decoding nonbinary QC-LDPC codes whose parity-check arrays consist of α-multiplied CPMs [2].

The $(16120, 15345)$ QC-LDPC code (or a shorten code) given in this paper is a good candidate for applications in high-speed optical communication systems (such as 400 Gbs Ethernet) and flash memory (or hard disk drivers) where a very low error rate is required, i.e., a well performing code with very low error-floor is required. Based on our extensive studies and simulations, we find that, to achieve a BER of 10^{-14} or below without an error-floor, the parity-check matrix of a high-rate (0.93 or above) LDPC code needs to have an average column weight at least 6. For a long high-rate LDPC code, if the average column weight of its parity-check matrix is 4, it will have an error-floor above the BER of 10^{-10} unless a pre-processing or a post-processing is included in its decoder (or being concatenated with an outer code). To achieve a BER of 10^{-13} without a visible error-floor, an average column weight 5 is needed. We did construct several high-rate high-performance QC-LDPC codes of length around 16k with rate 0.93 or above which achieve the BER of 10^{-15} without visible error-floors (presented at the conferences without giving specific constructions). The construction of the $(16120, 15345)$ QC-LDPC code given in this paper is specific and we strongly believe that the code will achieve the BER of 10^{-15} or below without an error-floor. The decoder complexity issue resulting from the large column weight will be resolved by the reduced-complexity iterative decoding scheme, CPM-RID, presented in this paper.

Also presented in this paper were two high-rate shortened codes of the $(16120, 15345)$ QC-LDPC code. They also perform well with the proposed CPM-RMSA and they can be decoded by the same decoder of the mother $(16120, 15345)$ QC-LDPC code.

Furthermore, a method for constructing $(3, 6)$-regular QC-LDPC codes of rate-$\frac{1}{2}$ with girth 8 was also proposed in this paper. The method is based on masking 4×8 submatrices of the base matrix $\mathbf{B}$ of the mother $(16120, 15345)$ QC-LDPC code.

The construction of the base matrix $\mathbf{B}$ for the $(16120, 15345)$ QC-LDPC code can be generalized by taking any two arbitrary subsets chosen from a given finite field and forming a matrix whose entries are the sum of the elements from the two chosen subsets of the given field.

Acknowledgment

This work was supported in part by the NSF under Grant CCF-1015548 and in part by gift grants from Northrop Grumman Corporation, LSI Corporation and SanDisk.

References

[1] R. G. Gallager, "Low Density Parity-check codes," *IRE Trans. Inform. Theory*, vol. IT-8, no. 1, pp. 21-28, Jan. 1962.

[2] W. E. Ryan and S. Lin, *Channel Codes: Classical and Modern*, New York: Cambridge Univ. Press, 2009.

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

[4] Y. Chen and K. Parhi, "Overlapped message passing for quasi-cyclic low-density parity check codes," *IEEE Trans. Circuits and Systems I*, vol. 51, no. 6, pp. 1106-1113, Jun. 2004.

[5] K. Liu, S. Lin and K. Abdel-Ghaffar, "A revolving iterative algorithm for decoding algebraic quasi-cyclic LDPC codes," *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Istanbul, Turkey, Jul. 7-12, 2013.

[6] K. Liu. S. Lin, and K. Abdel-Ghaffar, "A revolving iterative algorithm for decoding algebraic cyclic and quasi-cyclic LDPC codes," *IEEE Trans. Commun.*, vol. 61, no. 12, pp.4816–4827, Dec. 2013.

[7] S. Song, B. Zhou, S. Lin, and K. Abdel-Ghaffar, "A unified approach to the construction of binary and nonbinary quasi-cyclic LDPC codes based on finite fields," *IEEE Trans. Commun.*, vol. 57, no. 1, pp. 84-93, Jan. 2009.

[8] 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. Inform. Theory*, vol. 53, no. 7, pp. 2429–2458, Jul. 2007.

[9] Q. Diao, Q. Huang, S. Lin and K. Abdel-Ghaffar, "A matrix-theoretic approach for analyzing quasi-cyclic low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 58, no.6, pp. 4030–4048, Jun. 2012.

[10] M. Karimi and A. H. Banihashemi, "Counting short cycles of quasi cyclic protograph LDPC codes," *IEEE Commun. Lett.*, vol 16, no 3, pp.400–403, Mar. 2012.

[11] Q. Diao, Y.Y. Tai, S. Lin and K. Abdel-Ghaffar, "LDPC codes on partial geometries: construction, trapping set structure, and puncturing," *IEEE Trans. Inform. Theory*, vol. 59, no. 12, pp. 7898-7914, Dec. 2013.

[12] J. Chen and M. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 50, no. 3, pp. 406-414, Mar. 2002.

[13] J. Xu, L. Chen, I . Djurdjevic, S. Lin and K. Abdel-Ghaffar, "Construction of regular and irregular LDPC codes: geometry decomposition and masking," *IEEE Trans. Inform. Theory*, vol. 53, no. 1, pp. 121-134, Jan. 2007.

[14] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications. revised ed*. Cambridge, UK: Cambridge University Press, 1994.

[15] X.-Y Hu, E. Eleftheriou, and D.-M. Arnold, "Progressive edge-growth Tanner graphs," *Proc. IEEE GlobeCom*, pp. 995–1001, Nov. 2001.

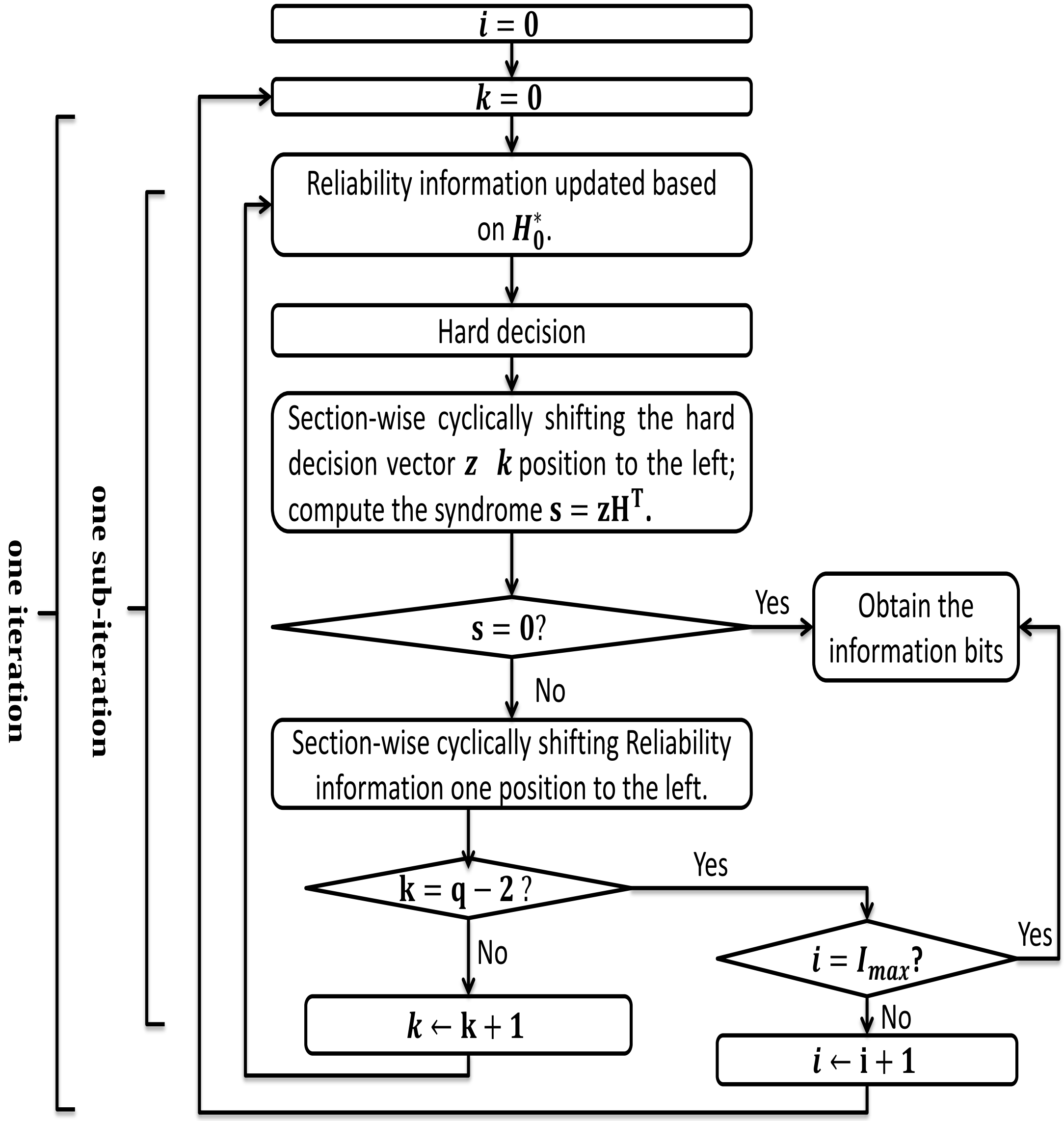
$i = 0$
$k = 0$
Reliability information updated based on H_0^*.
Hard decision
Section-wise cyclically shifting the hard decision vector z k position to the left; compute the syndrome $\mathbf{s} = \mathbf{z}\mathbf{H}^T$.
$\mathbf{s} = \mathbf{0}$?
Yes
Obtain the information bits
No
Section-wise cyclically shifting Reliability information one position to the left.
$\mathbf{k} = \mathbf{q} - 2$?
Yes
No
$i = I_{max}$?
Yes
No
$k \leftarrow \mathbf{k} + 1$
$i \leftarrow \mathbf{i} + 1$
one sub-iteration
one iteration

Fig. 1. A flow chart of the CPM-RID decoding scheme.

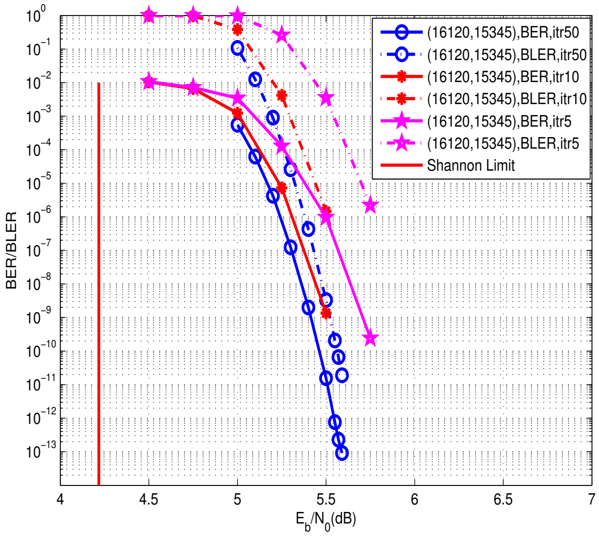

Fig. 2. Performance of the (16120,15345) QC-LDPC code decoded with a scaled MSA over AWGN channel.

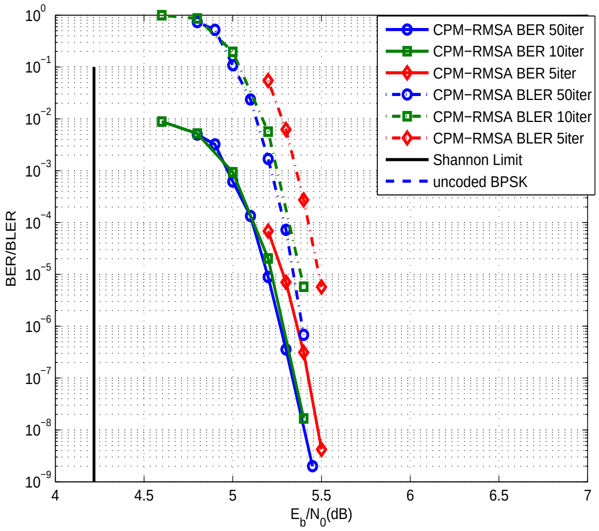

Fig. 3. Performance of the (16120,15345) QC-LDPC code decoded with the CPM-RMSA over AWGN channel.

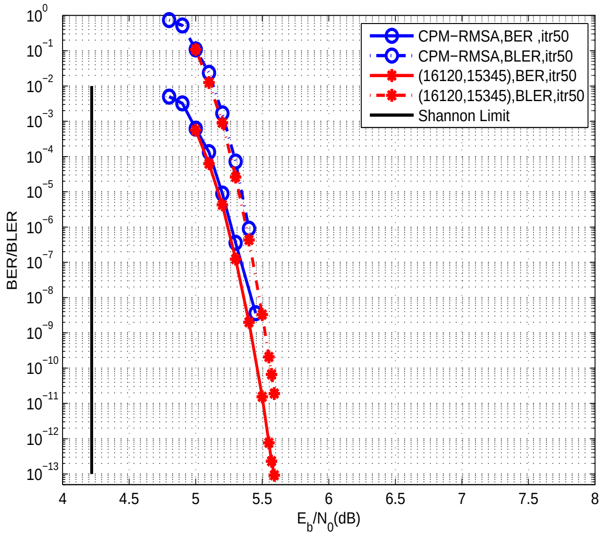

Fig. 4. Performance comparison of the (16120,15345) QC-LDPC code decoded with the CPM-RMSA and a scaled MSA.

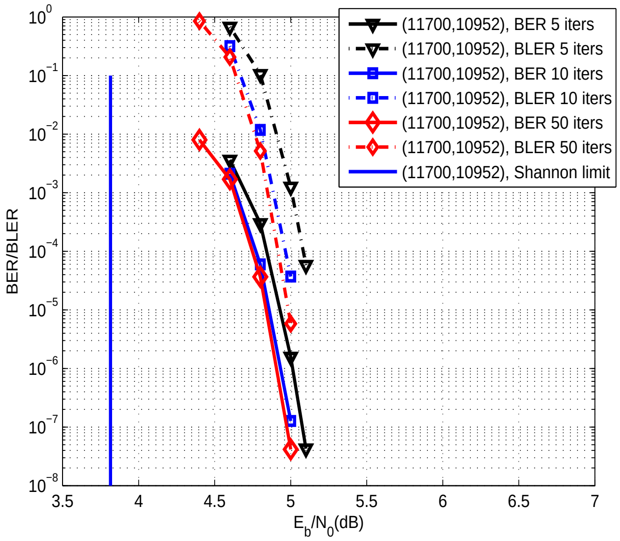

Fig. 5. Performance of the (11700, 10925) QC-LDPC code decoded with the CPM-RMSA over AWGN channel.

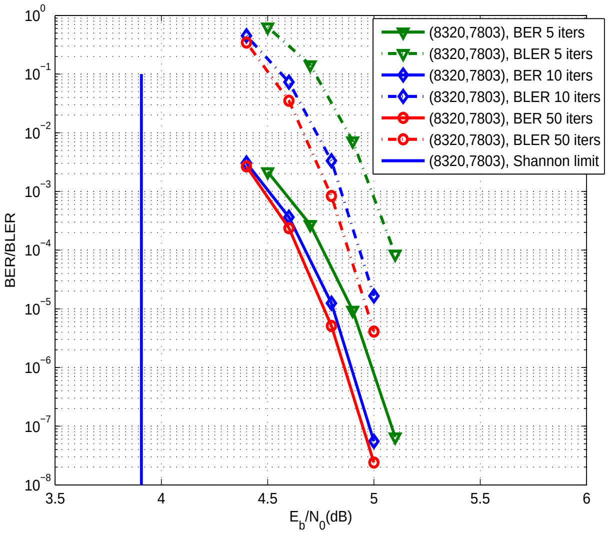

Fig. 6. Performance of the (8320, 7803) QC-LDPC code decoded with the CPM-RMSA over AWGN channel.

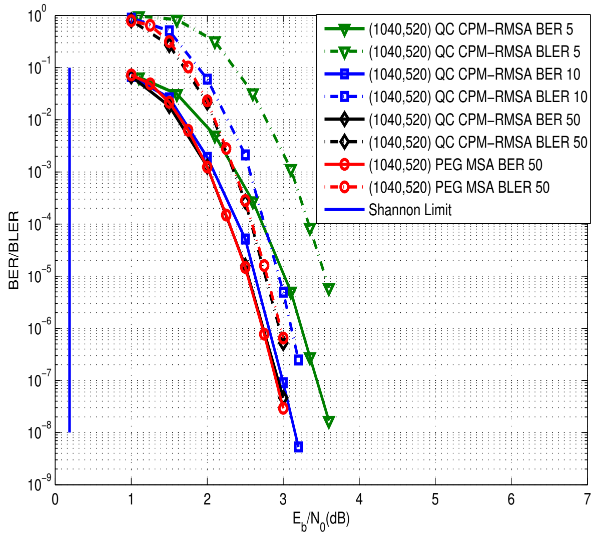

Fig. 7. Performance of the (1040, 520) QC-LDPC code decoded with the CPM-RMSA over AWGN channel.