TABLE II
THE SECOND SWD FOR SOME SMALL KASAMI CODES

$m=2$		$m=3$		$m=4$		$m=5$	
w	A_w	w	A_w	w	A_w	w	A_w
9	70	42	5 544	180	361 760	744	22 915 200
10	135	44	4 410	184	137 700	752	4 312 968
11	90	46	10 584	188	856 800	760	60 888 960
12	215	48	7 707	192	255 595	768	8 292 779
13	90	50	10 584	196	856 800	776	60 888 960
14	45	52	2 646	200	107 100	784	3 805 560
15	6	54	1 960	204	218 400	792	17 836 160

Theorem 1: The second support weight distribution of the $[2^{2m}-1, 3m, 2^{2m-1}-2^{m-1}]$ Kasami code is given by the expressions in Table I.

We have verified the second SWD for some small Kasami codes by computer, and these numbers are shown in Table II.

It appears to be more difficult to determine higher order support weight distributions completely. The most difficult case is probably when all the γ_i are distinct. For instance, studying a three-dimensional subcode, we have one nonzero word in V_{γ_1} and two words in each of three cosets $V_{\gamma_1}+\boldsymbol{c}(a_1,0)$, $V_{\gamma_1}+\boldsymbol{c}(a_2,0)$, and $V_{\gamma_1}+\boldsymbol{c}(a_1+a_2,0)$. Since only three out of the six coset points may be chosen freely, it is not obvious how to divine the weights of the remaining three. Maybe it can be done in combination with other methods.

REFERENCES

[1] T. Helleseth, T. Kløve, and J. Mykkeltveit, "The weight distribution of irreducible cyclic codes with block lengths $n_1((q^l-1)/n)$," *Discr. Math.*, vol. 18, pp. 179–211, 1977.

[2] V. K. Wei, "Generalized Hamming weights for linear codes," *IEEE Trans. Inf. Theory*, vol. 37, no. 5, pp. 1412–1418, Sep. 1991.

[3] O. Milenkovic, S. T. Coffey, and K. J. Compton, "The third support weight enumerators of the doubly-even, self-dual $[32, 16, 8]$ codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 3, pp. 740–746, Mar. 2003.

[4] S. Dougherty, A. Gulliver, and M. Oura, "Higher weights and graded rings for binary self-dual codes," *Discr. Appl. Math.*, vol. 128, pp. 251–261, 2003.

[5] H. G. Schaathun, "Duality and support weight distibutions," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 862–867, May 2004.

[6] T. Kløve, "Support weight distribution of linear codes," *Discr. Math.*, vol. 106/107, pp. 311–316, 1992.

[7] J. Simonis, "The effective length of subcodes," *Appl. Algebra Engrg. Comm. Comput.*, vol. 5, no. 6, pp. 371–377, 1994.

[8] T. Helleseth and P. V. Kumar, "The weight hierarchy of the Kasami codes," *Discr. Math.*, vol. 145, no. 1–3, pp. 133–143, 1995.

Quasi-Cyclic LDPC Codes for Fast Encoding

Seho Myung, Kyeongcheol Yang, *Member, IEEE*, and Jaeyoel Kim

***Abstract*—In this correspondence we present a special class of quasi-cyclic low-density parity-check (QC-LDPC) codes, called block-type LDPC (B-LDPC) codes, which have an efficient encoding algorithm due to the simple structure of their parity-check matrices. Since the parity-check matrix of a QC-LDPC code consists of circulant permutation matrices or the zero matrix, the required memory for storing it can be significantly reduced, as compared with randomly constructed LDPC codes. We show that the girth of a QC-LDPC code is upper-bounded by a certain number which is determined by the positions of circulant permutation matrices. The B-LDPC codes are constructed as irregular QC-LDPC codes with parity-check matrices of an almost lower triangular form so that they have an efficient encoding algorithm, good noise threshold, and low error floor. Their encoding complexity is linearly scaled regardless of the size of circulant permutation matrices.**

***Index Terms*—Block cycle, circulant permutation matrix, efficient encoding, low-density parity-check (LDPC) codes, quasi-cyclic codes.**

I. INTRODUCTION

Low-density parity-check (LDPC) codes—first discovered by Gallager [7] and rediscovered by Sipser *et al.* [13] and MacKay *et al.* [10], [11]—have created much interest recently since they are shown to have a remarkable performance close to the Shannon limit over additive white Gaussian noise (AWGN) channels [14]. LDPC codes possess many advantages including parallelizable decoding, self-error-detection capability by syndrome check, an asymptotically better performance than turbo codes, etc. Many coding theorists have brought new developments in the construction and decoding schemes of LDPC codes with low complexity for their commercial use in the past few years.

Richardson *et al.* introduced a method to design LDPC codes that perform extremely close to the Shannon capacity for sufficiently large code length under the assumption of no cycles [14]. They computed the threshold of noise level for a large class of binary-input channels by *density evolution*, and presented some simulation results for proving their claims. Here, the threshold of noise level means the maximum noise level to obtain the zero error probability as the block length tends to infinity. To find an ensemble which has larger threshold than that of the conventional ensemble of LDPC codes, Kasai *et al.* introduced *detailedly represented* irregular LDPC codes [8] and Richardson and Urbanke introduced *multi-edge type* LDPC codes [16]. They are obtained by representing the degree distribution according to the type of the edges.

Density evolution is a useful tool to obtain the asymptotical performance of LDPC codes, but not to estimate their performance in the case of finite length. In other words, it is not guaranteed that finite-length LDPC codes with degree distribution suggested by density evolution have good performance. The performance of LDPC codes of

Manuscript received April 27, 2004; revised October 23, 2004. This work was supported in part by the Center for Broadband OFDM Mobile Access (BrOMA) at the Pohang University of Science and Technology (POSTECH) supported by the ITRC program of the Korean Ministry of Information and Communication (MIC) under the supervision of the Institute of Information Technology Assessment (IITA).

S. Myung and K. Yang are with the Department of Electronics and Electrical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Kyungbuk 790-784, Korea (e-mail: kcyang@postech.ac.kr).

J. Kim is with Samsung Electronics Co., Ltd., Suwon, 416, Maetan-3 dong, Yeongtong-gu, Suwon, Gyeonggi, 442-742, Korea (kimjy@samsung.com).

Communicated by M. P. C. Fossorier, Associate Editor for Coding Techniques.

Digital Object Identifier 10.1109/TIT.2005.851753

finite length may be affected by other factors such as cycle property and minimum distance, etc. Therefore, finite-length analysis for LDPC codes over general channels may be a challenging problem. Di *et al.* [2] presented an analysis of finite-length LDPC codes when used over the binary erasure channel (BEC). However, their analysis may not be easily extended to the case of more general channels and, therefore, it is still an open problem to derive a finite-length analysis of LDPC codes over general channels.

Regardless of many advantages of LDPC codes, the encoding problem of LDPC codes may be an obstacle for their commercial applications because they have higher encoding complexity than turbo codes. Recently, many people have tried to solve it in some structured forms by taking not only the ensemble of codes with a parity-check matrix of a lower triangular shape but also cyclic or quasi-cyclic codes like the LDPC codes based on finite geometries [9], etc.

MacKay *et al.* [12] and Richardson *et al.* [15] showed that the encoding complexity is upper-bounded by $O(N) + O(g^2)$, where N is the code length and g is the *gap* to measure the "distance" between a given parity-check matrix and a lower triangular matrix. Therefore, it may be possible to reduce the encoding complexity if we can reduce the gap g. In the extreme case of $g = 0$, the corresponding LDPC codes have linear encoding complexity. The irregular repeat–accumulate (IRA) codes can be regarded as LDPC codes with $g = 0$. However, g is not generally required to be zero in order to get linear encoding complexity. This may be possible if the matrix ϕ of a given parity-check matrix which causes $O(g^2)$ encoding complexity in [15] is chosen to be a special form, say, the *identity matrix*.

Another disadvantage of general LDPC codes is that a significant amount of memory is needed to store their parity-check matrices. Quasi-cyclic LDPC (QC-LDPC) codes may be a good candidate to solve the memory problem, since their parity-check matrices consist of circulant permutation matrices or the zero matrix. In fact, the required memory for storing them can be reduced by a factor $1/L$, when $L \times L$ circulant permutation matrices are employed. A good example for QC-LDPC codes is the array codes [4].

In this correspondence, we present a special class of QC-LDPC codes, called block-type LDPC (B-LDPC) codes, which are linearly encodable and have better memory efficiency as compared with the conventional LDPC codes. A B-LDPC code is defined as an irregular QC-LDPC code whose parity-check matrix is an almost lower triangular matrix with an additional constraint that the corresponding matrix ϕ is the identity matrix. These constraints on the structure of its parity-check matrix guarantee that it can be linearly encodable regardless of the size of circulant permutation matrices. Furthermore, simulation results show that well-designed B-LDPC codes have no performance degradation due to the constraint on their special structure, as compared with randomly constructed LDPC codes.

The outline of the correspondence is as follows. In Section II, we review QC-LDPC codes and analyze their cycle structure. We discuss a simple condition under which QC-LDPC codes have cycles and give a result on the girth of QC-LDPC codes. In Section III, we propose B-LDPC codes for efficient encoding and good performance. They have linear encoding complexity by properly choosing the matrix ϕ and have good performance by controlling their cycle structure. We verify their performance by simulations in Section IV. Finally, we give concluding remarks in Section V.

II. Quasi-Cyclic LDPC Codes

In this section, we review QC-LDPC codes and analyze their cycle properties. Due to the special structure of their parity-check matrices, their cycles may be analyzed easily in an algebraic way. The *girth* which is the minimum length of cycles in their Tanner graph may be estimated by the exponents of circulant permutation matrices and *block-cycles*, which will be defined later in this section.

A. Definition of QC-LDPC Codes

A QC-LDPC code is characterized by the parity-check matrix which consists of small square blocks which are the zero matrix or circulant permutation matrices. Let P be the $L \times L$ permutation matrix given by

$$P = \begin{bmatrix} 0 & 1 & 0 & & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & 0 & & 0 \end{bmatrix}. \tag{1}$$

Note that P^i is just the circulant permutation matrix which shifts the identity matrix I to the right by i times for any integer i, $0 \le i < L$. For simple notation, P^∞ denotes the zero matrix.

Let $\boldsymbol{H}$ be the $mL \times nL$ matrix defined by

$$\boldsymbol{H} = \begin{bmatrix} P^{a_{11}} & P^{a_{12}} & \dots & P^{a_{1(n-1)}} & P^{a_{1n}} \\ P^{a_{21}} & P^{a_{22}} & \dots & P^{a_{2(n-1)}} & P^{a_{2n}} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ P^{a_{m1}} & P^{a_{m2}} & \dots & P^{a_{m(n-1)}} & P^{a_{mn}} \end{bmatrix} \tag{2}$$

where $a_{ij} \in \{0, 1, \dots, L-1, \infty\}$. The code $\mathcal{C}$ with parity-check matrix $\boldsymbol{H}$ is *quasi-cyclic* in the sense that $\boldsymbol{c} = (\boldsymbol{c}_0, \boldsymbol{c}_1, \dots, \boldsymbol{c}_{n-1}) \in \mathcal{C}$ implies that $\hat{T}^i \boldsymbol{c} \in \mathcal{C}$ for all i, $0 \le i \le L-1$, where

$$\hat{T}^i \boldsymbol{c} \triangleq (T^i \boldsymbol{c}_0, T^i \boldsymbol{c}_1, \dots, T^i \boldsymbol{c}_{n-1})$$

and

$$T^i \boldsymbol{c}_l \triangleq (c_{l,i}, c_{l,i\oplus 1}, \dots, c_{l,i\oplus L-1})$$

for $\boldsymbol{c}_l = (c_{l,0}, c_{l,1}, \dots, c_{l,L-1})$. Here $\oplus$ denotes the modulo-L addition.

From now on, $\mathcal{C}$ will be referred to as a *QC-LDPC code*. When $\boldsymbol{H}$ has full rank, then its code rate is given by

$$R = \frac{Ln - Lm}{Ln} = \frac{n-m}{n} = 1 - \frac{m}{n}$$

regardless of its code length $N = nL$. If the locations of 1's in the first row of the ith row block $\boldsymbol{H}_i \triangleq [P^{a_{i1}} \cdots P^{a_{in}}]$ are fixed, then the locations of other 1's in $\boldsymbol{H}_i$ are uniquely determined. Therefore, the required memory for storing the parity-check matrix of the QC-LDPC code can be reduced by a factor $1/L$, as compared with randomly constructed LDPC codes.

The QC-LDPC code may be regular or irregular depending on the choice of a_{ij}'s of $\boldsymbol{H}$ in (2). When $\boldsymbol{H}$ has no blocks corresponding to the zero matrix, it is a regular LDPC code with column weight m and row weight n. In this case, its code rate is larger than $1 - m/n$ since there exist at least $m - 1$ linearly dependent rows. Two examples of QC-LDPC codes are in the followings.

Example 1 (Array Code): For a prime q and a positive integer $j \le q$, the parity-check matrix of the array code is defined by

$$\boldsymbol{H}(q, j) \triangleq \begin{bmatrix} I & I & I & \dots & I \\ I & P & P^2 & \dots & P^{q-1} \\ I & P^2 & P^4 & \dots & P^{2(q-1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ I & P^{j-1} & P^{(j-1)2} & \dots & P^{(j-1)(q-1)} \end{bmatrix}. \tag{3}$$

Therefore, the array code is a QC-LDPC code with $L = q, n = q$, and $m = j$. Since each column of $\boldsymbol{H}(q, j)$ has j ones and each row has q ones, it can be regarded as a (j, q) regular LDPC code. It is easily shown that $\boldsymbol{H}(q, j)$ has rank $qj - j + 1$ and therefore its rate is given by

$$R = \frac{q^2 - qj + j - 1}{q^2} = 1 - \frac{qj - j + 1}{q^2} > 1 - \frac{j}{q}.$$

It is well-known that for $j \ge 3$, it has Tanner graph of girth 6 [4], [17]. □

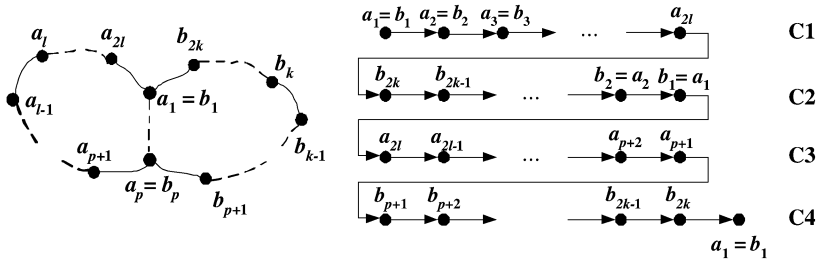

Fig. 1. p overlaps between two block-cycles and the corresponding chain.

Example 2 (Modified Array Code): For efficient encoding, Eleftheriou *et al.* [3] proposed a modified array code with the following parity-check matrix:

$$\boldsymbol{H}(q,j,k) \triangleq \begin{bmatrix} I & I & I & \dots & I & \dots & I \\ 0 & I & P & \dots & P^{(j-2)} & \dots & P^{(k-2)} \\ 0 & 0 & I & \dots & P^{2(j-3)} & \dots & P^{2(k-3)} \\ \vdots & \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 0 & I & \dots & P^{(j-1)(k-j)} \end{bmatrix}.$$

It is an irregular QC-LDPC code with $L = q$, $n = k$, $m = j$ where q is prime and $q \geq k \geq j$. Clearly, $\boldsymbol{H}(q,j,k)$ has full rank since it is an upper triangular matrix with nonzero diagonal elements. Therefore, its rate is given by

$$R = \frac{qk - qj}{qk} = \frac{k-j}{k} = 1 - \frac{j}{k}.$$

Due to the upper triangular form of $\boldsymbol{H}(q,j,k)$, it can be efficiently encoded. It is easily checked that there are no cycles of length 4 in the corresponding Tanner graph. □

B. The Cycle Properties of QC-LDPC Codes

Since the cycles of short length in the parity-check matrix of an LDPC code may degrade its performance, it is more critical to eliminate them. Furthermore, QC-LDPC codes always have a cycle of finite length under some constraints. In order to analyze these situations, we define a *block-cycle* and an overlap between block-cycles. Consider the $m \times n$ matrix $M(\boldsymbol{H})$ obtained from replacing zero matrices and circulant permutation matrices by "0" and "1," respectively, in the parity-check matrix $\boldsymbol{H}$ of a QC-LDPC code given in (2). $M(\boldsymbol{H})$ is called the *mother matrix* (or *base matrix*) of $\boldsymbol{H}$. If there is a cycle generated by these 1's in $M(\boldsymbol{H})$, it is called a *block-cycle*. If a circulant permutation matrix belongs to two or more block-cycles, it is called an overlap between these block-cycles.

A block-cycle of length $2l$ in $\boldsymbol{H}$ of (2) may be represented by the chain

$$P^{a_1} \rightarrow P^{a_2} \rightarrow \cdots \rightarrow P^{a_{2l}} \rightarrow P^{a_1}.$$

Here both P^{a_i} and P^{a_i+1} are located in either the same column block or the same row block of $\boldsymbol{H}$, and both P^{a_i} and P^{a_i+2} are located in the distinct column blocks and row blocks. This block-cycle is simply called a $2l$-*block-cycle*.

Proposition 3: Let $P^{a_1} \rightarrow P^{a_2} \rightarrow \cdots \rightarrow P^{a_{2l}} \rightarrow P^{a_1}$ be the chain corresponding to a $2l$-block-cycle. If r is the least positive integer such that

$$r \cdot \sum_{i=1}^{2l} (-1)^{i-1} a_i \equiv 0 \bmod L, \tag{4}$$

then the block-cycle leads to a cycle of length $2lr$.

Proof: Without loss of generality, we assume that P^{a_1} and P^{a_2} are located in the same row block and consider the "1" of P^{a_1} at the jth row in the row block. Clearly, the "1" of P^{a_2} at the same row in the row block is located at the $(j + a_2)$th column of P^{a_2}. Since P^{a_2} and P^{a_3} are located in the same column block, the "1" at the $(j + a_2)$th column of P^{a_3} is located at the $(j + a_2 - a_3)$th row of P^{a_3}. Continuing this procedure through the block-cycle, it is easily checked that the "1" at the jth row of P^{a_1} is connected to the "1" at the $(j + a_2 - a_3 + \cdots + a_{2l} - a_1)$th row of P^{a_1} in the block-cycle. Repeating this procedure $r - 1$ times, the "1" at the jth row of P^{a_1} in the row block is connected to the "1" at the $\left(j + r \sum_{i=1}^{2l} (-1)^i a_i\right)$th row of P^{a_1} in the row block. This procedure may be represented by the following chain:

$$P^{a_1} \rightarrow P^{a_2} \rightarrow \cdots \rightarrow P^{a_{2l}} \rightarrow \underbrace{\cdots}_{(r-1) \text{ repetitions}} \rightarrow P^{a_1}.$$

By the assumption of (4), we have

$$j \equiv j + r \cdot \sum_{i=1}^{2l} (-1)^i a_i \bmod L$$

for any integer j. This implies that there should be a cycle of length $2lr$. □

Fossorier, in [5], [6], presented essentially the same results on the necessary and sufficient condition under which there are cycles in the QC-LDPC codes. By Proposition 3, it may be possible to express the cycles of QC-LDPC codes into simple equations and therefore avoid the cycles of small length by choosing the exponents of circulant permutation matrices appropriately.

When we wish to construct a good QC-LDPC code, it seems very hard to find its parity-check matrix having no cycles. If the total number of 1's in its parity-check matrix increases in order to improve its performance, more block-cycles and overlaps may appear inevitably. The following theorem gives an upper bound on the minimum length of cycles in the QC-LDPC codes under the existence of an overlap between two block-cycles.

Theorem 4: If there are p overlaps between a block-cycle of length $2l$ and a block-cycle of length $2k$ in a QC-LDPC code, then there exists a cycle of length $2(2l+2k-p)$ which is twice the number of the distinct blocks in these block-cycles. Furthermore, the girth of the QC-LDPC code is at most $2(2l + 2k - p)$.

Proof: Assume that there are p overlaps between a block-cycle of length $2l$ and a block-cycle of length $2k$ and consider the chain of the permutation matrices, as shown in Fig. 1. Assign a_1 and a_p to the starting point and the ending point of overlaps, respectively. Therefore, $a_j = b_j$, for all $j = 1, 2, \ldots, p$. Without loss of generality, we assume that P^{a_1} and P^{a_2} are located in the same column block. Then P^{a_1}, $P^{a_{2l}}$, and $P^{b_{2k}}$ are located in the same row block by the definition of a block-cycle. In the same manner, P^{a_p}, $P^{a_{p+1}}$, and $P^{b_{p+1}}$ are located in the same row block or column block.

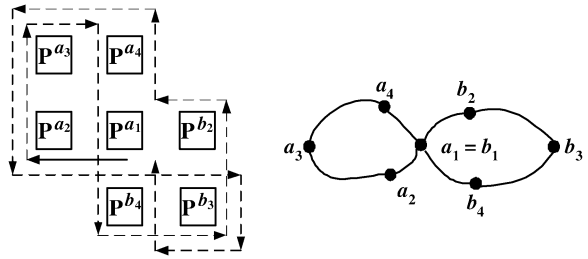

Fig. 2. An overlap between two block-cycles.

For simplicity, the chain can be divided into four subchains $C1, C2, C3, C4$, as shown in Fig. 1. Then it is easily checked that each subchain contributes the following:

$$\begin{aligned} C1 &\to \sum_{i=1}^{2l} (-1)^{i-1} a_i \\ C2 &\to \sum_{i=0}^{2k-1} (-1)^i b_{2k-i} \\ C3 &\to \sum_{i=0}^{2l-(p+1)} (-1)^i a_{2l-i} \\ C4 &\to \sum_{i=p+1}^{2k} (-1)^{i-1} b_i, \end{aligned}$$

and the chain leads to the following equation:

$$\sum_{i=1}^{2l} (-1)^{i-1} a_i + \sum_{i=0}^{2k-1} (-1)^i b_{2k-i} + \sum_{i=0}^{2l-(p+1)} (-1)^i a_{2l-i} + \sum_{i=p+1}^{2k} (-1)^{i-1} b_i \equiv 0 \bmod L$$

regardless of a_i, b_i, and L. Therefore, there is a cycle of length

$$2l + 2k + (2l - p) + (2k - p) = 2(2l + 2k - p)$$

by Proposition 3, regardless of L and the exponents of the permutation matrices. □

Corollary 5: Assume that there are no zero matrices in (2). Then the girth of the corresponding regular QC-LDPC codes is at most 12 for any $m \geq 2$ and $n \geq 3$.

Proof: In a regular QC-LDPC code having no zero matrices there are always two block-cycles of length 4 with an overlap consisting of two blocks. Therefore, the girth is at most 12 by Theorem 4. □

Corollary 5 is a well-known fact derived by Fossorier [5], [6], but our approach gives another simple proof on it. Moreover, Theorem 4 gives more results on the girth of QC-LDPC codes, as shown in the following examples.

Example 6: Consider a QC-LDPC code whose parity-check matrix has an overlap between two block-cycles of length 4 in Fig. 2. Proposition 3 tells us that the chain of length 14 given by

$$\begin{aligned} &P^{a_1} \to P^{a_2} \to P^{a_3} \to P^{a_4} \to P^{b_4} \to P^{b_3} \to P^{b_2} \to \\ &P^{a_1} \to P^{a_4} \to P^{a_3} \to P^{a_2} \to P^{b_2} \to P^{b_3} \to P^{b_4} \to P^{a_1} \end{aligned}$$

forms a cycle of length 14 regardless of L, a_i's, and b_i's in the block-cycle, since

$$a_1 - a_2 + a_3 - a_4 + b_4 - b_3 + b_2 - a_1 + a_4 - a_3 + a_2 - b_2 + b_3 - b_4 \equiv 0 \bmod L.$$

Therefore, the girth of the QC-LDPC code is at most 14 regardless of the size and the exponents of circulant permutation matrices. □

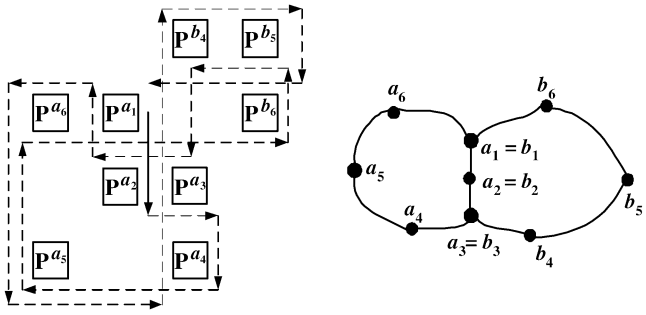

Fig. 3. Three overlaps between two block-cycles.

Example 7: Consider a QC-LDPC code whose parity-check matrix has three overlaps between two block-cycles of length 6 in Fig. 3. Proposition 3 tells us that the chain of length 18 given by

$$\begin{aligned} &P^{a_1} \to P^{a_2} \to P^{a_3} \to P^{a_4} \to P^{a_5} \to P^{a_6} \to \\ &P^{b_6} \to P^{b_5} \to P^{b_4} \to P^{a_3} \to P^{a_2} \to P^{a_1} \to \\ &P^{a_6} \to P^{a_5} \to P^{a_4} \to P^{b_4} \to P^{b_5} \to P^{b_6} \to P^{a_1} \end{aligned}$$

forms a cycle of length 18 regardless of L, a_i's, and b_i's in the block-cycle, since

$$a_1 - a_2 + a_3 - a_4 + a_5 - a_6 + b_6 - b_5 + b_4 - a_3 + a_2 - a_1 + a_6 - a_5 + a_4 - b_4 + b_5 - b_6 \equiv 0 \bmod L.$$

Therefore, the girth of the QC-LDPC code is at most 18 regardless of the size and the exponents of circulant permutation matrices. □

As in the array codes [4], [17], the cycles of regular QC-LDPC codes with algebraic constraints are more structurized than those of irregular QC-LDPC codes. For example, the number of cycles of minimum length in the array codes can be computed in the following.

Theorem 8: For any prime $q \geq 3$ and $q \geq j \geq 3$, the array code defined by $\boldsymbol{H}(q, j)$ in (3) has Tanner graph of girth 6. In particular, the number of the cycles of length 6 in the array code is given by

$$2q \binom{q}{2} \binom{j}{3}.$$

Proof: From [4], [17], it is well known that the girth of the array code is 6. For a cycle of length 6 in the array codes, the corresponding permutation matrices must form a block-cycle of length 6. The block-cycles of length 6 in the array code are one of the following six patterns:

$$\begin{bmatrix} P^{x_1a_1} & P^{x_1a_2} & \\ & P^{x_2a_2} & P^{x_2a_3} \\ P^{x_3a_1} & & P^{x_3a_3} \end{bmatrix}, \quad \begin{bmatrix} P^{x_1a_1} & P^{x_1a_2} & \\ P^{x_2a_1} & & P^{x_2a_3} \\ & P^{x_3a_2} & P^{x_3a_3} \end{bmatrix}$$

$$\begin{bmatrix} & P^{x_1a_2} & P^{x_1a_3} \\ P^{x_2a_1} & P^{x_2a_2} & \\ P^{x_3a_1} & & P^{x_3a_3} \end{bmatrix}, \quad \begin{bmatrix} P^{x_1a_1} & & P^{x_1a_3} \\ P^{x_2a_1} & P^{x_2a_2} & \\ & P^{x_3a_2} & P^{x_3a_3} \end{bmatrix}$$

$$\begin{bmatrix} P^{x_1a_1} & & P^{x_1a_3} \\ & P^{x_2a_2} & P^{x_2a_3} \\ P^{x_3a_1} & P^{x_3a_2} & \end{bmatrix}, \quad \begin{bmatrix} & P^{x_1a_2} & P^{x_1a_3} \\ P^{x_2a_1} & & P^{x_2a_3} \\ P^{x_3a_1} & P^{x_3a_2} & \end{bmatrix}$$

where $0 \leq a_1 < a_2 < a_3 \leq q - 1$ and $0 \leq x_1 < x_2 < x_3 \leq j - 1$. Without loss of generality, we assume that x_1, x_2, x_3, a_1, and a_2 are fixed in the first pattern. Then we have

$$x_1a_1 + x_2a_2 + x_3a_3 \equiv x_1a_2 + x_2a_3 + x_3a_1 \bmod q$$

that is,

$$a_1(x_1 - x_3) + a_2(x_2 - x_1) \equiv a_3(x_2 - x_3) \bmod q.$$

Since q is prime and $x_2 \neq x_3 \bmod q$, a_3 is uniquely determined. The same arguments are applied to the other patterns. Therefore, the number of cycles of length 6 in the array code is

$$q \times \frac{\binom{q}{2}\binom{j}{3} \times 6}{3} = 2q\binom{q}{2}\binom{j}{3}.$$

Here, the factor 3 in the denominator appears since each case is counted three times. □

III. Block-Type LDPC Codes for Efficient Encoding

Based on the analysis of the structure of QC-LDPC codes in the previous section, we present irregular QC-LDPC codes with parity-check matrices of an almost lower triangular form for efficient encoding and good performance. In fact, LDPC codes with parity-check matrices of an almost lower triangular form were considered for efficient encoding by MacKay *et al.* [12], Richardson *et al.* [15], and Eleftheriou *et al.* [3]. The QC-LDPC codes proposed in this section can be encoded easily by using a modified Richardson–Urbanke method [15]. In addition, some of their information or parity bits in these codes can be punctured to support high rate.

For convenience, we divide the parity-check matrix $\boldsymbol{H}$ in (2) into two parts: the information part $\boldsymbol{H}_{\rm I}$ and the parity part $\boldsymbol{H}_{\rm P}$, i.e., $\boldsymbol{H} = [\boldsymbol{H}_{\rm I} \quad \boldsymbol{H}_{\rm P}]$. For efficient encoding, we restrict the parity part $\boldsymbol{H}_{\rm P}$ of the parity-check matrix to an almost lower triangular matrix with additional constraints. Consider the following $mL \times (m+k)L$ parity-check matrix:

$$\begin{aligned}\boldsymbol{H} &= [\boldsymbol{H}_{\rm I}|\boldsymbol{H}_{\rm P}] \\ &= \begin{bmatrix} & P^{b_1} & I & 0 & \dots & 0 & 0 \\ & 0 & P^{b_2} & I & \dots & 0 & 0 \\ & \vdots & 0 & P^{b_3} & \dots & 0 & 0 \\ \boldsymbol{H}_{\rm I} & P^{y} & \vdots & \vdots & \dots & \vdots & \vdots \\ & \vdots & \vdots & \vdots & \dots & I & 0 \\ & 0 & 0 & 0 & \dots & P^{b_{m-1}} & I \\ & P^{x} & 0 & 0 & \dots & 0 & P^{b_m} \end{bmatrix}\end{aligned} \tag{5}$$

where P is the $L \times L$ circulant permutation matrix given in (1) and P^y is located in the lth row block for an integer $l \neq 1, m$. Note that l is conventionally chosen as about the half of m, even though l can be arbitrarily selected. The code $\mathcal{C}$ with parity-check matrix $\boldsymbol{H}$ given in (5) will be referred to as a block-type LDPC code, or *B-LDPC code* for short.

Based on the Richardson–Urbanke encoding method $\boldsymbol{H}$ is divided into the form

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} & \boldsymbol{T} \\ \boldsymbol{C} & \boldsymbol{D} & \boldsymbol{E} \end{pmatrix} \tag{6}$$

where $\boldsymbol{A}$ is $(m-1)L \times kL$, $\boldsymbol{B}$ is $(m-1)L \times L$, $\boldsymbol{T}$ is $(m-1)L \times (m-1)L$, $\boldsymbol{C}$ is $L \times kL$, $\boldsymbol{D} = P^x$ is $L \times L$, $\boldsymbol{E}$ is $L \times (m-1)L$, and $n = m + k$.

From now on, we assume that $\boldsymbol{H}_{\rm P}$ has full rank, i.e., the rows of $\boldsymbol{H}_{\rm P}$ are linearly independent. Using the Gaussian elimination, it is easily checked that the rows of $\boldsymbol{H}_{\rm P}$ are linearly independent if $\phi := \boldsymbol{E}\boldsymbol{T}^{-1}\boldsymbol{B} + \boldsymbol{D}$ is nonsingular.

Let $\boldsymbol{c}$ be a codeword of the code specified by $\boldsymbol{H}$, that is, $\boldsymbol{H}\boldsymbol{c}^T = 0^T$. This splits into two equations as follows:

$$\begin{aligned}\boldsymbol{A}\boldsymbol{s}^T + \boldsymbol{B}\boldsymbol{p}_1^T + \boldsymbol{T}\boldsymbol{p}_2^T &= 0 \\ (\boldsymbol{E}\boldsymbol{T}^{-1}\boldsymbol{A} + \boldsymbol{C})\boldsymbol{s}^T + (\boldsymbol{E}\boldsymbol{T}^{-1}\boldsymbol{B} + \boldsymbol{D})\boldsymbol{p}_1^T &= 0\end{aligned}$$

where $\boldsymbol{s}$ denotes the systematic part, $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ denote the parity parts which have length L and $(m-1)L$, respectively. Then we can obtain $\boldsymbol{p}_1$ as follows:

$$\boldsymbol{p}_1^T = \phi^{-1}(\boldsymbol{E}\boldsymbol{T}^{-1}\boldsymbol{A} + \boldsymbol{C})\boldsymbol{s}^T = \phi^{-1}(\boldsymbol{E}\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{s}^T + \boldsymbol{C}\boldsymbol{s}^T).$$

Because the matrix ϕ^{-1} is not sparse in general, the overall complexity of computing $\boldsymbol{p}_1$ is $O(N) + O(L^2)$ where $N(= nL)$ is the code length [15]. But, if ϕ can be chosen as the *identity matrix*, then the encoding complexity may be linearly scaled. The key idea of our algorithm for efficient encoding is to choose the matrix ϕ as the identity matrix (or a simple circulant permutation matrix in general) by a suitable selection of P^x and P^y in (5). Therefore, the overall complexity of computing $\boldsymbol{p}_1$ can be reduced to $O(N)$ regardless of the size of circulant permutation matrices.

In order to compute ϕ, consider $\boldsymbol{H}$ given in (5) and (6). It is easily shown that $\boldsymbol{T}^{-1}$ can be written as the matrix at the bottom of the page, where

$$P^{(i,j)} \triangleq \prod_{k=i}^{j} P^{b_k} = P^{b_i + b_{i+1} + \cdots + b_j}.$$

Since

$$\boldsymbol{B}^T = [\,(P^{b_1})^T \quad 0 \quad \dots \quad (P^{y})^T \quad 0 \quad \dots \quad 0\,]$$

and

$$\boldsymbol{E} = [\,0 \quad 0 \quad \dots \quad 0 \quad P^{b_m}\,]$$

we have

$$\begin{aligned}\boldsymbol{E}\boldsymbol{T}^{-1}\boldsymbol{B} &= P^{b_m}[P^{(2,m-1)}P^{(3,m-1)}\cdots P^{b_m-1}I]\boldsymbol{B} \\ &= P^{b_m}P^{(2,m-1)}P^{b_1} + P^{b_m}P^{(l+1,m-1)}P^{y} \\ &= P^{(1,m)} + P^{(l+1,m)}P^{y}\end{aligned}$$

where P^y is located in the lth row block of $\boldsymbol{B}$. Therefore, a proper choice for x and y can be summarized in the following theorem.

Theorem 9: Let $\boldsymbol{H}$ be given in (5) and (6). If x and y are chosen such that

$$x \equiv \sum_{i=1}^{m} b_i \bmod L \text{ and } y \equiv -\sum_{i=l+1}^{m} b_i \bmod L \tag{7}$$

or

$$\sum_{i=1}^{m} b_i \equiv 0 \bmod L \text{ and } x \equiv y + \sum_{i=l+1}^{m} b_i \bmod L \tag{8}$$

then the matrix $\phi \triangleq \boldsymbol{E}\boldsymbol{T}^{-1}\boldsymbol{B} + \boldsymbol{D}$ becomes the identity matrix.

As a result, the detailed operations for encoding B-LDPC codes can be summarized as follows.

$$\boldsymbol{T}^{-1} = \begin{bmatrix} I & 0 & 0 & \dots & 0 & 0 & 0 \\ P^{b_2} & I & 0 & \dots & 0 & 0 & 0 \\ P^{(2,3)} & P^{b_3} & I & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & 0 & 0 \\ P^{(2,m-2)} & P^{(3,m-2)} & P^{(4,m-2)} & \dots & P^{b_{m-2}} & I & 0 \\ P^{(2,m-1)} & P^{(3,m-1)} & P^{(4,m-1)} & \dots & P^{(m-2,m-1)} & P^{b_{m-1}} & I \end{bmatrix}$$

TABLE I
COMPUTATIONAL COMPLEXITY OF EACH STEP IN THE ENCODING

Step	Number of the required XOR operations
Step 1)	$cN-(3m+1)L$
Step 2)	$(m-2)L$
Step 3)	L
Step 4)	mL
Total	$(c-1+R)N-2L$

Encoding Procedure for B-LDPC Codes

- Step 1) Compute $\boldsymbol{A}\boldsymbol{s}^T$ and $\boldsymbol{C}\boldsymbol{s}^T$.
- Step 2) Compute $\boldsymbol{E}\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{s}^T=[P^{(2,m)}\cdots P^m]\boldsymbol{A}\boldsymbol{s}^T$.
- Step 3) Compute $\boldsymbol{p}_1^T$ by $\boldsymbol{p}_1^T=\boldsymbol{E}\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{s}^T+\boldsymbol{C}\boldsymbol{s}^T$.
- Step 4) Compute $\boldsymbol{p}_2^T$ by $\boldsymbol{T}\boldsymbol{p}_2^T=\boldsymbol{A}\boldsymbol{s}^T+\boldsymbol{B}\boldsymbol{p}_1^T$.

The computational complexity at each step in the encoding procedure is shown in Table I, where R is the code rate, $N=nL$ is the code length, and c is the average of column weights in $\boldsymbol{H}$. The complexity for the cyclic-shift operation within a block of length L is assumed to be negligible, so we count only the XOR operation (i.e., modulo-2 addition). Let $w(X)$ be the weight of a binary matrix X, that is, the number of 1's in X. Since $w(\boldsymbol{A})+w(\boldsymbol{C})=cN-(2m+1)L$, the number of the required operations at Step 1) is $cN-(3m+1)L$. The number of required operations at Step 2) is $(m-2)L$, since $w(\boldsymbol{E}\boldsymbol{T}^{-1})=(m-1)L$. Clearly, we need only L operations at Step 3). It is also easily checked that we need $2L$ operations for computing $\boldsymbol{A}\boldsymbol{s}^T+\boldsymbol{B}\boldsymbol{p}_1^T$ and $(m-2)L$ operations for computing $\boldsymbol{p}_2^T$ by back-substitution at Step 4), since $w(\boldsymbol{B})=2L$. Therefore, the number of the required operations for encoding B-LDPC codes is given by $(c-1+R)N-2L$, where $R=1-m/n$.

Assuming that each submatrix of a parity-check matrix with an almost lower triangular form has the same weight as that of $\boldsymbol{H}$ for a B-LDPC code in the form (6) and that ϕ^{-1} has weight $L^2/2$, it is easily checked that $(c-1+R)N-3L+L^2/2$ operations are required for encoding it by Richardson–Urbanke method. If we define ρ by

$$\rho=\frac{(c-1+R)N-2L}{(c-1+R)N-3L+L^2/2}=\frac{(c-1+R)n-2}{(c-1+R)n-3+L/2}$$

then $\rho<1$ whenever $L>2$. Note that ρ is smaller for larger L. Therefore, the complexity for encoding B-LDPC codes gets smaller and smaller than that for encoding general LDPC codes with parity-check matrix of an almost lower triangular form, when L increases.

Remark: For a simpler construction, we usually set $P^{b_i}=I$ for all $i=1,2,\ldots,m-1$. Then the conditions (7) and (8) in Theorem 9 may be simply expressed as follows:

$$x\equiv b_m \bmod L \text{ and } y\equiv -b_m \bmod L$$

or

$$b_m\equiv 0 \bmod L \text{ and } x\equiv y \bmod L.$$

The B-LDPC codes constructed in this way have a simple structure and a fast encoding. One remaining issue here is how to make B-LDPC codes have good performance in this approach. □

Principle $\mathcal{A}$ for Constructing Good B-LDPC Codes: As a first step, we fix the column blocks with low weight and assign proper circulant permutation matrices to these column blocks so that the minimum length of the cycles between the nodes of low degree is maximized. Next, we allocate appropriate circulant permutation matrices to the column blocks corresponding to the nodes of high degree so that the girth of the code is maximized. □

An extensive simulation shows that the B-LDPC codes designed under Principle $\mathcal{A}$ have good bit-error rate (BER) and frame-error rate (FER) performance. In particular, they have a lower error floor, as compared with B-LDPC codes constructed without considering Principle $\mathcal{A}$. These experiments recommend that Principle $\mathcal{A}$ should be taken into account when we construct good LDPC codes with or without any structure. It has been believed that the nodes of high degree are robust to errors in the channel since they have more paths for updating the messages, whereas the nodes of low degree are weak to the errors because they have fewer paths for updating the messages [1], [14]. Thus, it may be guessed that the nodes of low degree are affected too frequently by the nodes with lower reliability if they are linked to a short cycle with nodes of low degree.

Based on Principle $\mathcal{A}$, our construction procedure for good B-LDPC codes is summarized as follows.

Construction Procedure for Good B-LDPC Codes

- Step 1) Obtain a good degree distribution for B-LDPC codes with m row blocks and $n(=m+k)$ column blocks by density evolution. Let f_i be the fraction of the variable nodes connected to exactly i check nodes. Then we have $f_2=\frac{m-1}{n}$ and $f_i=\frac{p_i}{n}$ for i, $3\le i\le m$, regardless of the size of the circulant permutation matrices. Here, p_i is a nonnegative integer such that $p_3\ge 1$ and
$$\sum_{i=2}^{m} f_i=1.$$
- Step 2) Set the parity part $\boldsymbol{H}_{\mathrm{P}}$ of the parity-check matrix as shown in (5), and choose P^x and P^y satisfying (7) or (8) such that ϕ is the identity matrix.
- Step 3) Combined with the results in Steps 1) and 2), construct an $m\times n$ mother matrix M so that it has as few block-cycles of short length and overlaps between them as possible.
- Step 4) Initialize $\boldsymbol{H}=[\,0\quad \boldsymbol{H}_{\mathrm{P}}\,]$, where 0 is the zero matrix of size $mL\times kL$.
- Step 5) For each "1" in the degree-3 columns of the mother matrix, replace the corresponding block of $\boldsymbol{H}$ by P^i for i, $0\le i\le L-1$, and list the girth and the number of the shortest cycles in the replaced matrix.
- Step 6) Among all the cases in Step 5), select the position of "1" in the mother matrix and the circulant permutation matrix such that the corresponding girth is maximized and then the number of the shortest cycles is minimized. Update $\boldsymbol{H}$ by adding the selected circulant permutation matrix at the selected position.
- Step 7) Repeat Steps 5) and 6) until each "1" in the degree-3 columns of the mother matrix is assigned to a circulant permutation matrix.
- Step 8) Repeat the same procedures as Steps 5), 6), and 7) for each degree $j>3$ in turn. □

When we obtain a good degree distribution for B-LDPC codes by density evolution in Step 1), it is not required to take care of the average error probability of the parity bits in most cases. Rather, we are interested only in the average error probability of the information bits. That is, our goal for construction of B-LDPC codes is to obtain degree distributions which achieve the zero error probability of the information bits. In addition, the mother matrix selected in Step 3) can be easily justified by Theorem 4, but its realization is done by search.

IV. SIMULATION RESULTS

Computer simulations were done to analyze the performance of B-LDPC codes obtained by our construction procedure over an AWGN

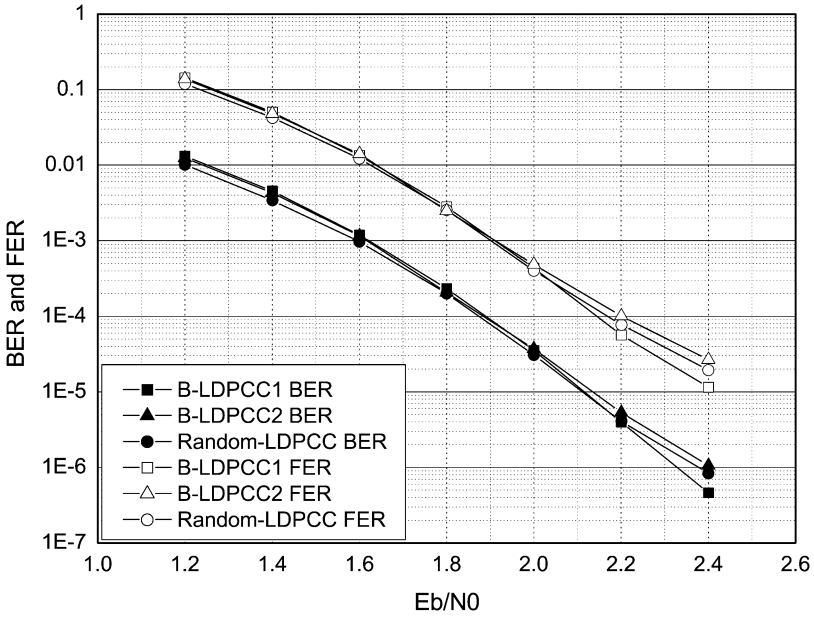

Fig. 4. Performance comparison between B-LDPC codes and a randomly constructed LDPC code with rate $1/2$ and length 1000.

channel. They were encoded efficiently according to the encoding procedure in Section III.

Fig. 4 shows the performance of two half-rate B-LDPC codes of length 1000, compared with that of a randomly constructed LDPC code with the same rate and length in terms of BER and FER. The codes denoted by B-LDPCC1 and B-LDPCC2 are B-LDPC codes in (5) and the code Random-LDPCC is a randomly constructed irregular LDPC code. Their parity-check matrices have almost the same degree distribution, that is, the variable node fractions of B-LDPCC1 and B-LDPCC2 are $f_2 = 0.475$, $f_3 = 0.350$, $f_9 = 0.175$, and those of Random-LDPCC are $f_2 = 0.491$, $f_3 = 0.335$, $f_9 = 0.174$. Their corresponding noise thresholds are $\sigma^* = 0.951$ and $\sigma^* = 0.954$, respectively. Note that Random-LDPCC has the best known degree distribution obtained by Richardson *et al.* [14] under the constraint with maximum variable node degree 9.

Both B-LDPCC1 and Random-LDPCC are designed under Principle $\mathcal{A}$, while B-LDPCC2 is constructed without considering it. In both cases of B-LDPCC1 and Random-LDPCC, their girth is 6 and their minimum length of the cycles between degree-2 and/or degree-3 nodes is 12. On the other hand, B-LDPCC2 has girth 6 and its minimum length of the cycles between degree-2 and/or degree-3 nodes is 6. The parity-check matrices for B-LDPCC1 and B-LDPCC2 have 20×40 blocks of size 25×25.

In each case, we let the decoder run for enough iterations to get the best possible performance. The performance of B-LDPCC1 is almost the same as that of Random-LDPCC at low signal-to-noise ratio (SNR), but the former is better than the latter at high SNR. B-LDPCC2 appears to have an error floor at high SNR. Therefore, the B-LDPC codes designed under Principle $\mathcal{A}$ have no performance degradation due to the constraint on their special structure, as compared with randomly constructed LDPC codes.

V. Concluding Remarks

We discussed how to construct B-LDPC codes for fast encoding and efficient storage. First, we analyzed the cycle properties of QC-LDPC codes, gave a condition under which they have a cycle, and showed that their girth is upper-bounded by a certain number which is determined by the positions of circulant permutation matrices. Second, we proposed B-LDPC codes for fast encoding. They have a fast encoding algorithm with linear complexity due to the structure of their parity-check matrices regardless of size of circulant permutation matrices. Furthermore, the required memory for storing their parity-check matrices can be reduced in inverse proportion to the size of circulant permutation matrices.

In order to construct good B-LDPC codes, we found a good degree distribution by density evolution and designed their parity-check matrices under Principle $\mathcal{A}$. Simulation results show that the proposed B-LDPC codes have no performance degradation due to the constraint on their special structure, as compared with randomly constructed LDPC codes.

Acknowledgment

The authors wish to thank the three reviewers for their valuable comments which greatly improved the presentation of these results.

References

[1] S.-Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. dissertation, MIT, Cambridge, MA, 2000.

[2] C. Di, D. Proietti, İ. E. Telatar, T. J. Richardson, and R. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1570–1579, Jun. 2002.

[3] E. Eleftheriou and S. Olcer, "Low-density parity-check codes for multilevel modulation," in *Proc. IEEE Int. Symp. Information Theory (ISIT2002)*, Lausanne, Switzerland, Jun./Jul. 2002, p. 442.

[4] J. L. Fan, "Array codes as low-density parity-check codes," in *Proc. 2nd Int. Symp. Turbo Codes*, Brest, France, Sep. 2000, pp. 543–546.

[5] M. P. C. Fossorier, "Quasi-cyclic low density parity check codes," in *Proc. 2003 IEEE Int. Symp. Information Theory (ISIT2003)*, Yokohama, Japan, Jun./Jul. 2003, p. 150.

[6] ——, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1788–1794, Aug. 2004.

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

[8] K. Kasai, T. Shibuya, and K. Sakaniwa, "Detailed representation of irregular LDPC code ensembles and density evolution," in *Proc. IEEE Int. Symp. Information Theory (ISIT2003)*, Yokohama, Japan, Jun./Jul. 2003, p. 121.

[9] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometires: A rediscovery and new results," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.

[10] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[11] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low-density parity-check codes," *Electron. Lett.*, vol. 32, pp. 1645–1646, Aug. 1996.
[12] D. J. C. MacKay, S. T. Wilson, and M. C. Davey, "Comparison of constructions of irregular Gallager codes," *IEEE Trans. Commun.*, vol. 47, no. 10, pp. 1449–1454, Oct. 1999.
[13] M. Sipser and D. A. Spielman, "Expander codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1710–1722, Nov. 1996.
[14] T. J. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
[15] T. J. Richardson and R. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 638–656, Feb. 2001.
[16] ——, "Multi-edge type LDPC codes," *IEEE Trans. Inf. Theory*. Available: [Online]. http://lthcwww.epfl.ch/, to be published.
[17] K. Yang and T. Helleseth, "On the minimum distance of array codes as LDPC codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 12, pp. 3268–3271, Dec. 2003.

Dual MacWilliams Pair

Dae San Kim, *Member, IEEE*

***Abstract*—A pair of posets (P, Q) on $[n]$ is called a weak dual MacWilliams pair $(wdMp)$ if the P-weight enumerator of a linear code uniquely determines the Q-weight enumerator of the dual of that code for every linear code of length n over a finite field. First, we show that $(P, \check{P})$ is a $wdMp$ if and only if the group of all P-weight preserving linear automorphisms of the ambient n-dimensional space over the finite field acts transitively on every P-sphere centered at 0. Here $\check{P}$ is the dual poset of P. Also, we show some equivalent conditions which say that P being weak order poset with $Q = \check{P}$ is essentially the only possible case for (P, Q) to be a $wdMp$.**

***Index Terms*—MacWilliams-type identity, (weak) dual MacWilliams pair, P-weight enumerator, weak order poset.**

I. Introduction and Statement of the Main Theorems

The poset-codes were introduced in [1] by Brualdi *et al.* in connection with Niederreiter's problem in [9] and have received considerable attention in recent years.

Let $\mathbb{F}_q$ be the finite field with q elements, and let P be a poset on the underlying set $[n] = \{1, 2, \ldots, n\}$ of coordinate positions of vectors in $\mathbb{F}_q^n$. Then the P-weight $w_P(u)$ for $u \in \mathbb{F}_q^n$ is defined as the cardinality of the smallest ideal containing the support of u (cf. (6)). Then $d_P(u, v) = w_P(u - v)$ is a metric, called P-metric. If P is an antichain, P-weight and P-metric reduce, respectively, to Hamming weight and Hamming metric.

As in the case of "classical" MacWilliams identity, one may seek to find a MacWilliams-type identity for the P-weight enumerator of a linear code C of length n over $\mathbb{F}_q$, which encodes the information about P-weight distribution $\{A_{P,i}(C)\}_{i=0}^n$ of C. Here

$$A_{P,i}(C) = |\{u \in C \mid w_P(u) = i\}|. \tag{1}$$

Manuscript received April 6, 2004; revised December 9, 2004. This work was supported by the Basic Research Program of the Korea Science and Engineering Foundation under Grant R01-2002-000-00083-0(2004).

The author is with the Department of Mathematics, Sogang University, Seoul 121-742, Korea (e-mail: dskim@sogang.ac.kr).

Communicated by A. E. Ashikhmin, Associate Editor for Coding Theory.
Digital Object Identifier 10.1109/TIT.2005.851765

Some previous but not quite successful results in this direction are [3], [4], and [6].

Recently, when $P = n_1\mathbf{1} \oplus n_2\mathbf{1} \oplus \cdots \oplus n_t\mathbf{1}$ (cf. (10)) is a weak order poset, the desired MacWilliams-type identity was found in [7] by Kim and Oh and also in [5] independently by the present author. However, the methods employed in [5] and [7] are quite different.

In [7], the P-weight enumerator for a linear code C is defined as

$$W(C, P \mid x) = \sum_{u \in C} x^{w_P(u)} = \sum_{i=0}^{n} A_{P,i}(C) x^i. \tag{2}$$

The discrete Poisson summation formula is applied in order to find a MacWilliams-type identity for the leveled P-weight enumerator $W(C, P \mid x; y_0, y_1, \ldots, y_t)$, containing the auxiliary variables $y_0, y_1, \ldots, y_t$ and reducing to $W(C, P \mid x)$ in (2) for $y_0 = y_1 = \cdots = y_t = 1$. Then the desired identity, expressing $W(C^\perp, \check{P} \mid x)$ in terms of $W(C, P \mid x)$ ($\check{P}$ is the dual poset of P), follows by specializing the auxiliary variables to be 1.

On the other hand, in [5] the P-weight enumerator (called there a sphere enumerator) is defined as the linearized version of (2). Namely

$$S(C, P \mid x_0, \ldots, x_n) = \sum_{u \in C} x_{w_P(u)} = \sum_{i=0}^{n} A_{P,i}(C) x_i. \tag{3}$$

The result obtained in [5] is

$$S(C^\perp, \check{P} \mid \mathfrak{X}) = \frac{1}{|C|} S(C, P \mid \widetilde{\Theta}\mathfrak{X}) \tag{4}$$

where $\mathfrak{X} = {}^t(x_0, x_1, \ldots, x_n)$ and $\widetilde{\Theta}$ is an explicit invertible matrix. Here $\widetilde{\Theta}$ can be viewed as a generalization of the Krawtchouk matrix. The idea of proof is to consider first the fragment enumerator $F(C, P \mid Z)$ containing finer information about the code C than $S(C, P \mid \mathfrak{X})$, and to find the identity

$$F(C^\perp, \check{P} \mid Z) = \frac{1}{|C|} F(C, P \mid \Theta Z). \tag{5}$$

Here, again, Θ is an explicit invertible matrix and (5) is obtained by using the discrete Poisson summation formula, the Möbius inversion formula of Rota, and a suitable linear change of variables (this is really the crux of the matter) corresponding to Θ. Then "collecting terms" of the coefficients of $F(C, P \mid Z)$ and of the entries of Θ, we obtain the desired identity in (4).

So, when P is a weak order poset, in both [5] and [7] the P-weight enumerator of C uniquely determines the $\check{P}$-weight enumerator of $C^\perp$. This fact was proved in [8] and [10] for the special case of a weak order poset P with $n_1 = n_2 = \cdots = n_t = 1$ (the $n = 1$ case of Rosenbloom–Tsfasman metric on $Mat_{n,s}(\mathbb{F}_q)$, cf. [2], [10]). Kim and Oh [7] went further by showing that P is a weak order poset if the P-weight enumerator of C uniquely determines the $\check{P}$-weight enumerator of $C^\perp$.

Motivated by this, for P, Q posets on $[n]$ we call (P, Q) a weak dual MacWillams pair $(wdMp)$ if the P-weight enumerator of C uniquely determines the Q-weight enumerator of $C^\perp$ and it is a dual MacWilliams pair (dMp) if (P, Q) and (Q, P) both are $wdMp$'s.

Our first main result is the following theorem.

Theorem A: The following are equivalent.

1) P is a weak order poset on $[n]$.
2) $(P, \check{P})$ is a $wdMp$.
3) The group $\mathrm{Aut}\left(\mathbb{F}_q^n, w_p\right)$ (cf. (8)) acts transitively on each P-sphere $S_P(r)$ for $0 \le r \le n$ (cf. (7)).

1) $\Rightarrow$2) is shown in [5] and [7] and 2) $\Rightarrow$ 1) in [7], as we mentioned earlier. 1) $\Rightarrow$ 3) is proved in [5] and here we show 3) $\Rightarrow$ 1). 3) $\Rightarrow$ 2) is alluded to in [2] and [10] (see, especially, [10, p. 325, lines 6–14]) for the special case of a weak order poset P with $n_1 = n_2 = \cdots = n_t = 1$.