WP1-A-4

A Parity Check Matrix Design for Irregular LDPC Codes with 2K Block Length

Chutima Prasartkaew* and Somsak Choomchuay†

*College of Data Storage Technology and Applications, King Mongkut's Institute of Technology Ladkrabang, BKK, Thailand
Tel:/Fax: + 66-2-326-4731, E-mail: prasartkaew@yahoo.com

†Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, BKK 10520, Thailand
Tel: +66-2-326-4222 Ext.114, Fax: +66-2-739-2398, E-mail: kchsomsa@kmitl.ac.th

Abstract— **This paper outlines the work on another design of a parity check matrix for Irregular LDPC codes. The design is based on the pattern of Modified Array and Interleaved Modified Array LDPC codes. The application of matrix transposition Quasi-cyclic shifting has resulted in the reduction of 1's. The designed matrix is suitable for codes with short and medium block lengths. The code rate of 0.56 at the BER of** 10^{-4} **is obtained.**

I. INTRODUCTION

Error Correction Codes (ECC) is one of many tools made available for achieving consistent data transmission. It can improve bit error rate (BER) in alternative to increasing the signal to noise ratio (SNR). Two major type of ECCs known as block code and convolution codes have played their good roles in many applications, in particular modern communication and data storage technology. Low-Density Parity-Check codes (LDPC) are also linear block code that has been studied vastly in this decade. The main advantage of the codes is that they provide the performance at that very close to the capacity for a lot of different channels and linear time complex algorithms for decoding [12]. They also suit well the parallel realization.

In this paper we propose a modified method to obtain parity check matrices. The obtained result is comparable to a published work [1] but with a higher code rate. The rest of this paper is organized as follows: General perspective of LDPC codes is given in section II. Encoder and decoder are included. A design of parity check matrices is outline in section III and the corresponding performance tests are reported in section IV.

II. LDPC CODES

Of its discovery in 1960 by Gallager [2], the LDPC code has been ignored for some ten years. This was because the code itself is quite complex. In the same time the more highly-structured code; Reed Solomon code, was introduced [3]. The introduction of Turbo code by Berrou, Glavieux and Thitimajshima in 1993 [4] has drawn great attention since the code performance is close to Shannon limit. LDPC codes was recovered in 1998 by Rechardson and Urbanke [5] and in 1999 by Mackay and Neal [6]. Since then, LDPC became more popular and widely developed for wider area of applications including communications and data storage.

According to their parity check matrix, LDPC codes can be termed as, random parity check matrix LDPC and structured parity check matrix LDPC. A random parity check matrix LDPC can have better BER compared to its competitor. However, it hold more complex parity check matrix. Many works have focused on the efficient design of a structured one [8, 9, 10].

There are two different ways to represent LDPC codes; matrix representation and graphical representation. In the matrix representation, as it is named, LDPC codes hold small number of "1" in each row and column, i.e. $W_c << n$ and $W_r << m$ for a dimension $m \times n$ parity matrix. This can provide large minimum distance of the code. However such a circumstance results a large parity check matrix. In the graphical representation, Tanner graph [7] provides an efficient view of LDPC codes. There are m check nodes (c-nodes; number of parity bits) and n variable nodes (v-nodes; number of bits in a codeword).

LDPC codes are said to be regular if W_c is constant for every column, and $W_r = W_c(n/m)$. If **H** is low density but the number of "1" in each row or column are not constant, the code is said to be an irregular one.

A. Encoder

Similar to all other linear block codes, we have the relation;

$$\mathbf{C}_{(1\times n)}\mathbf{H}^{\mathrm{T}}_{(n\times m)} = \mathbf{0} \tag{1}$$

$$\mathbf{C}\mathbf{H}^{\mathrm{T}} = [\mathbf{B}\,|\,\mathbf{P}]\begin{bmatrix}\mathbf{H}_1^{\mathrm{T}}\\ \mathbf{H}_2^{\mathrm{T}}\end{bmatrix} = \mathbf{B}\mathbf{H}_1^{\mathrm{T}} + \mathbf{P}\mathbf{H}_2^{\mathrm{T}} = \mathbf{0} \tag{2}$$

or

$$\mathbf{P} = \mathbf{B}\mathbf{H}_1^{\mathrm{T}} + \left(\mathbf{H}_2^{\mathrm{T}}\right)^{-1} \tag{3}$$

Where **C** is a codeword matrix, and **H** is a parity check matrix. $\mathbf{H}_2$ is a parity check matrix of the dimension $(m \times m)$. It is a part of the matrix **H**.

The task of the encoder is then to compute the matrix **P** that can be directly appended to the message to produce the codeword.

For the matrix **H** to be more manageable, LU decomposition method is preferably applied; i.e. $[\mathbf{H}]=[\mathbf{L}][\mathbf{U}]$. Thus,

$$\begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \underbrace{\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}}_{Y} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (4)$$

Let $[\mathbf{Y}]=[\mathbf{U}][\mathbf{P}]$, then we can use forward substitution to solve $[\mathbf{L}][\mathbf{Y}]=[\mathbf{B}]$.

$$\begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (5)$$

Finally, use backward substitution to solve $[\mathbf{U}][\mathbf{P}]=[\mathbf{Y}]$. There we can get $\{p_i\}$ as need.

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (6)$$

B. Decoder

The decoding algorithm used for LDPC codes was discovered independently and comes under different names. The most common are the belief propagation algorithm, the message passing algorithm and the sum-product algorithm.

Tanner graph is an intuitive way in understanding the LDPC decoder. The graph can be drawn directly from the **H** matrix as shown below:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad (7)$$

The graph contains *m* check nodes (number of parity bits) and *n* variable nodes (number of bits in a codeword). Check node f_i is connected to a variable node c_i if the element h_{ij} of **H** is a 1.

In the Log-domain Sum-Product algorithm, the message passes between check nodes and variable nodes. In each pass the log likelihood ratio is recorded for its probability of its likely symbol.

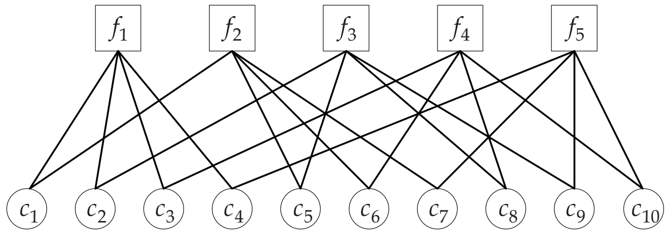

Fig. 1. Tanner graph of the H matrix given in (7)

III. DESIGNS OF PARITY CHECK MATRICES

The structure of **H** matrix can have great effect to an encoder, decoder and code performance. A design of the parity check matrix outlined in this paper is based on the modified array LDPC and interleave-modified array LDPC.

A. Some Previous Works

Fan [8] has introduced the array structure parity matrix that can offer comparable performance when compared to a random generated parity matrix. Other features are: low noise floor and no existence of cycle of 4. Fan's matrix is shown below.

$$\mathbf{H}(\mathrm{p},\mathrm{j},\mathrm{k}) \triangleq \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} \\ \mathbf{I} & \boldsymbol{\alpha} & \boldsymbol{\alpha}^2 & \cdots & \boldsymbol{\alpha}^{\mathrm{k}-1} \\ \mathbf{I} & \boldsymbol{\alpha}^2 & \boldsymbol{\alpha}^4 & \cdots & \boldsymbol{\alpha}^{2(\mathrm{k}-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{I} & \boldsymbol{\alpha}^{\mathrm{j}-1} & \boldsymbol{\alpha}^{2(\mathrm{j}-1)} & \cdots & \boldsymbol{\alpha}^{(\mathrm{j}-1)(\mathrm{k}-1)} \end{bmatrix} \quad (8)$$

This yields the code rate of $R=1-\frac{pj-j+1}{p^2}$. Where **I** is an identity matrix $(p\times p)$, codeword is a $k\times p$ size, parity bit is a $j\times p$ size and $\boldsymbol{\alpha}$ is a permutation matrix $(p\times p)$ representing a single left or right cyclic shift. For example, for $p=5$, we can have **I,** $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^2$ as shown below.

$$\mathbf{I} = \begin{bmatrix} 10000 \\ 01000 \\ 00100 \\ 00010 \\ 00001 \end{bmatrix}_{(5\times5)} \quad \boldsymbol{\alpha} = \begin{bmatrix} 01000 \\ 00100 \\ 00010 \\ 00001 \\ 10000 \end{bmatrix}_{(5\times5)} \quad \boldsymbol{\alpha}^2 = \begin{bmatrix} 00100 \\ 00010 \\ 00001 \\ 10000 \\ 01000 \end{bmatrix}_{(5\times5)}$$

Eleftheriou and Olcer [9] have proposed the modified array structure (MAC) by applying cyclic shift to Fan's array. This structure (shown in Eq. (9)) yields the code rate of $R=1-(j/k)$. MAC has superior performance to Fan's array that it can reduce the 1's in the lower triangle. The obtained matrix can also utilize a simple encoder. Efficient encoding is achieved from **H** without the need to compute the generator matrix of the code. As the upper triangular form of **H**, there are no cycles of length 4 in the corresponding Tanner graph.

$$\mathbf{H}=\begin{bmatrix} \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} & \mathbf{I} & \cdots & \cdots & \mathbf{I} \\ \mathbf{0} & \mathbf{I} & \boldsymbol{\alpha} & \cdots & \boldsymbol{\alpha}^{(j-2)} & \boldsymbol{\alpha}^{(j-1)} & \cdots & \boldsymbol{\alpha}^{(k-2)} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \cdots & \boldsymbol{\alpha}^{2(j-3)} & \boldsymbol{\alpha}^{2(j-2)} & \cdots & \boldsymbol{\alpha}^{2(k-3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{I} & \boldsymbol{\alpha}^{(j-1)} & \cdots & \boldsymbol{\alpha}^{(j-1)(k-j)} \end{bmatrix} \quad (9)$$

Where **0** is the $p \times p$ null matrix.

Singhaudom *et. al.* [10] has proposed the interleaved modified array LDPC or IMAC by introducing the quasi-cyclic matrix into the cyclic shift of [9]. This interleaved LDPC of which the parity matrix given below in Eq. (10) is superior to Fan's array LDPC when the block length is particularly long. The parameter is the circular shift permutation matrix.

$$\mathbf{H}=\begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I}\boldsymbol{\omega} & \mathbf{I}\boldsymbol{\omega}^2 & \mathbf{I}\boldsymbol{\omega}^3 & \cdots & \mathbf{I}\boldsymbol{\omega}^j \\ \mathbf{0} & \mathbf{I} & \boldsymbol{\alpha\omega} & \boldsymbol{\alpha}^2\boldsymbol{\omega}^2 & \boldsymbol{\alpha}^3\boldsymbol{\omega}^3 & \cdots & \boldsymbol{\alpha}^{(k-2)}\boldsymbol{\omega}^j \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \boldsymbol{\alpha}^2\boldsymbol{\omega}^2 & \boldsymbol{\alpha}^4\boldsymbol{\omega}^3 & \cdots & \boldsymbol{\alpha}^{2(k-3)}\boldsymbol{\omega}^j \\ \vdots & \vdots & \vdots & \mathbf{I} & \boldsymbol{\alpha}^3\boldsymbol{\omega}^3 & \cdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} & \cdots & \boldsymbol{\alpha}^{(j-1)}\boldsymbol{\omega}^{(k-j)} \end{bmatrix} \quad (10)$$

Where $\boldsymbol{\omega}$ is the $p \times p$ permutation matrix ($\boldsymbol{\omega}^n = \boldsymbol{\omega}$) representing a single left or right quasi-cyclic shift. For example, for $p = 5$, we can have $\mathbf{I}, \boldsymbol{\omega}$ and $\boldsymbol{\omega}^n = \boldsymbol{\omega}$ as shown below.

$$\mathbf{I}=\begin{bmatrix}10000\\01000\\00100\\00010\\00001\end{bmatrix}_{(5\times5)} \quad \boldsymbol{\omega}=\begin{bmatrix}01000\\00010\\10000\\00100\\00001\end{bmatrix}_{(5\times5)} \quad \boldsymbol{\omega}^5=\boldsymbol{\omega}=\begin{bmatrix}01000\\00010\\10000\\00100\\00001\end{bmatrix}_{(5\times5)}$$

$$\boldsymbol{\alpha}\times\boldsymbol{\omega}=\begin{bmatrix}01000\\00100\\00010\\00001\\10000\end{bmatrix}_{(5\times5)}\times\begin{bmatrix}01000\\00010\\10000\\00100\\00001\end{bmatrix}_{(5\times5)}=\begin{bmatrix}00010\\10000\\00100\\00001\\01000\end{bmatrix}_{(5\times5)}$$

B. Our Construction

In our work, the structured parity check matrix was designed aiming at a fairly simple check matrix. To get the suitable arbitrary parity check matrices, a good approach is to avoid constructing **H** matrix at all [11].

In our construction, the first half of **H** was designed by taking the left-hand half of the IMAC. The second half is primarily formed by transposition of the first half. There are 4 possibilities for the next efforts; 1) the matrix in the second half was cyclically shifted, 2) the matrix in the second half was quasi-cyclically shifted, 3) the matrix in the second half was flipped horizontally and cyclically shifted is applied to both halves, and 4) the matrix in the second half was flipped horizontally and quasi-cyclically shifted is applied to both halves. Option 3) works well compared to others [13]. Further modification to option 3) can be done by applying quasi-cyclic shifting. The final obtained matrix is shown in Eq. (11) below.

$$\mathbf{H}=\begin{bmatrix} \mathbf{I} & \boldsymbol{\omega} & \boldsymbol{\omega}^2 & \boldsymbol{\omega}^3 & \cdots & \boldsymbol{\omega}^{j-1} & \boldsymbol{\omega}^j \\ \mathbf{0} & \mathbf{I} & \boldsymbol{\alpha\omega} & \boldsymbol{\alpha}^2\boldsymbol{\omega}^2 & \cdots & \boldsymbol{\alpha}^{(k-2)}\boldsymbol{\omega}^j & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \boldsymbol{\alpha}^2\boldsymbol{\omega}^2 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \mathbf{I} & \cdots & \cdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} & \cdots & \mathbf{0} \end{bmatrix} \quad (11)$$

The matrix is also a triangular one with the dimension of $jp \times kp$. We still have the code rate of $R = 1-(j/k)$ where p is prime number and $j \le k \le p$, **0** is the $p \times p$ null matrix, and $\boldsymbol{\omega}$ is the $p \times p$ quasi cyclic matrix.

IV. Performance Evaluation

Our preliminary investigation was focused to the short and medium block length. Parameters are also designed to maintain the code rate of greater than 0.5. Test parameters are given in table 1 and table 2 below. The proposed matrix offers considerably good result compared to the existing LDPC codes.

For high BER, it can be observed that the parameter p is big compared to the parameter j. Then the sets of parameters which gives the highest BER and require block lengths (of ~1,000 bits for short and ~2,000 bits for medium block lengths) was selected and simulated for performance evaluation.

TABLE 1: PARAMETERS FOR SHORT BLOCK TESTING

j	3	4	5
k	11	11	11
p	97	97	97
$R=1-(j/k)$	0.727	0.636	0.545
$b=p(k-j)$	776	679	582
$Parity=jp$	291	388	485
$c=kp$	1,067	1,067	1,067
Iterations	5, 10, 20		

TABLE 2: PARAMETERS FOR MEDIUM BLOCK TESTING

j	5	6	7
k	16	16	16
p	137	137	137
$R=1-(j/k)$	0.688	0.625	0.563
$b=p(k-j)$	1,507	1,370	1,233
$Parity=jp$	685	822	959
$c=kp$	2,192	2,192	2,192
Iterations	5, 10, 20		

We firstly compared our design to the existing regular design [1] of the similar code rate. The code rate proposed by Rakibul *et. al.* [1] is 0.5 whilst ours is 0.563. The obtained result is shown in Fig. 1. It is clear that our code offers better performance, in particular when SNR is less than 6.5 dB.

When compared to existing irregular MAC and IMAC, our design also offers better performance. This is in particular when SNR is less 9 dB as shown in Fig.2.

We also investigated number of iterations versus the performance obtained. Our LDPC codes were tested at 5, 10, and 20 iterations. At 5 iterations, BER of both short and medium block length are similar. BER can be improved when we increased the number of iterations to 10. When the number of iteration is of 20, only the BER of a medium block length can be slightly improved. One may note that the obtained BER is still not so low. This is quite common that one cannot get so low BER when the block length is not large.

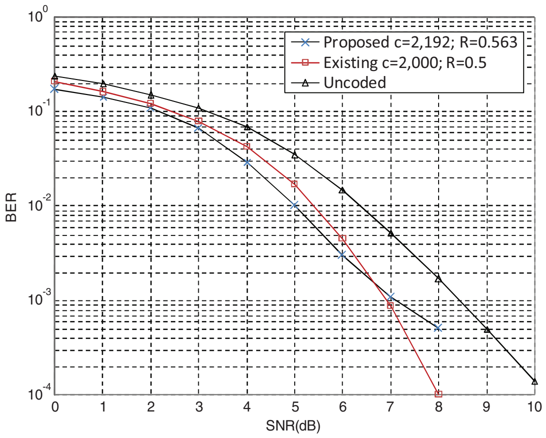

Fig. 2. Performance of our design compared to existing regular LDPC

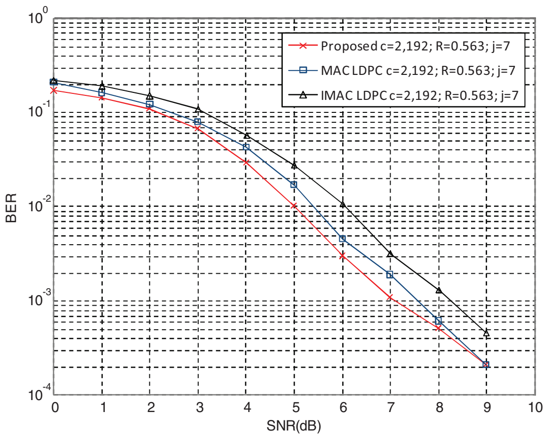

Fig. 3. The proposed LDPC and the existing LDPC's performance

V. CONCLUSIONS

We have proposed a design of parity check matrix for LDPC code based on the concept of array, modify array and interleaved modify array LDPC codes. The design is suitable for short and medium block length. The resulted design offers better performance compared to the regular design one. When comparing to the existing MAC and IMAC, our design also gives better performance. For the long block length, the obtained LDPC cannot compete the existing the interleaved modified array. Based on our idea, there could be opportunity to develop a parity check matrix on obtain the performance of IMAC for longer block length but with less complexity.

REFERENCES

[1] Mohammad Rakibul, Jinsang Kim, "On the use of QC-LDPC code for data transfer using short and medium block length," Conference ICACT 2009 Feb. 15-18, 2009.

[2] R. Gallager, "Low-density Parity-check Code," IRE Trans. Information Theory, Jan. 1962, pp.21-28.

[3] Reed, I. S. and Solomon, G., "Polynomial Codes Over Certain Finite Fields," *SIAM Journal of Applied Math.*, vol. 8, 1960, pp. 300-304.

[4] C Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting Coding and Decoding," proc. IEEE int. Conf., pp. 1064-1070, May 1993.

[5] T.J. Richardson, M.A. Shokrollahi, and R.L. Urbanke, "Design of Capacity-approaching Low-density Parity-check Codes," IEEE Trans. on Info. Theory, Vol. 47, pp. 619-637, Feb. 2001.

[6] D.J.C. Mackay and R. Neal, "Near Shannon Limit Performance of Low Density Parity Check Code," Electronics Letter, Vol.33, Mar 1997, pp.457-458.

[7] R.M. Tanner, "A Recursive Approach to Low Complexity Code," IEEE Trans. Information Theory, Sept. 1981, pp.533-547.

[8] J.L. Fan, "Array Codes as low-density parity-check codes," Proc. 2nd Int. Symp. Turbo Code, Beit, France, pp. 543-546, Sept. 2000.

[9] E. Eleftheriou and S.Olcer, "Low-density parity-Check Codes for Digital Subscriber Lines," Proc. 2002 Int. Conf. on Comm., pp. 1752-1757, April-May 2002.

[10] W. Singhaudom, S. Noppankeepong, P. Suphithi, "Design of Hign-Rate Modified Array Codes for Magnetic Recording System," ECTI International Conference, May 2007.

[11] O. Othman Khalifa, S. Khan, M. Zaid, and M. Nawawi, "Performance Evaluation of Low Density Parity Check Codes," International Journal of Computer Science and Engineering, 2008.

[12] W. E. Ryan, "An Introduction to LDPC Codes," *in CRC Handbook for Coding and Signal Processing for Recording Systems* (B. Vasic, ed.) CRC Press, 2004.

[13] Chutima Prasartkaew and Somsak Choomchuay, "A Design of Parity Check Matrix for Irregular LDPC Codes," ISCIT International Conference, Korea, Sept. 2009.