Abstract—In this letter, we present a linear-complexity encoding algorithm for any cycle GF($2^p$) code $C_E(G, H)$. We just need to investigate the case where $G$ is a nontrivial connected graph. If $G$ is a tree, the only codeword is the all-zero word. If $G$ is not a tree, first, we show that through graph analysis $H$ can be transformed into an equivalent block-diagonal upper-triangular form simply by permuting the rows and columns of $H$; then, we show that whether $H$ is full row-rank or not, the code can be encoded in linear time.

Index Terms—Cycle code, encoding algorithm, Galois fields.

I. INTRODUCTION

Allaguer’s binary low-density parity-check (LDPC) codes [1] are excellent error-correcting codes with performance close to the Shannon Capacity [2]. LDPC codes over GF($2^p$) have been investigated empirically by Davey and MacKay [3] over the binary-input AWGN channel. LDPC codes of column weight $j = 2$ are known as cycle codes [4]. Though distance properties of cycle codes are not as good as LDPC codes of column weight $j \geq 3$ [1], reference [5] shows that cycle GF($2^p$) codes have better performance than other LDPC codes, including degree-distribution-optimized binary irregular LDPC codes. Reduced complexity algorithms for decoding GF($2^p$) LDPC codes have been proposed in [6], [7]. Hence, cycle GF($2^p$) codes are promising in many applications.

The high encoding complexity prevents the application of cycle GF($2^p$) codes. Assume the block length of a cycle GF($2^p$) code is $n$, the conventional encoding method has complexity $O(n^2)$ [8]. For binary cycle codes, an efficient encoding algorithm with complexity $O(n)$ has been reported in [9]. The proposed algorithm removes one check node from the code’s Tanner graph [10] and spreads the rest in a structure called “pseudo-tree” [9]. It takes advantage of the natural property of binary cycle codes—their associated parity check matrices are not full row-rank. However, this does not hold for cycle GF($2^p$) codes. In this letter, we prove that cycle GF($2^p$) codes can be actually encoded in linear time. Our method works for both binary and nonbinary cycle codes.

Section II presents a graphical representation of cycle GF($2^p$) codes. Section III first presents the main theorem and then gives a complete proof after introducing some lemmas. The conclusion is given in Section IV.

Manuscript received January 8, 2006. The associate editor coordinating the review of this letter and approving it for publication was Prof. Marc Fossorier.

The authors are with the Dept. of Electronic Engineering and Information Science, University of Science and Technology of China, P.R.China (e-mail: fredandy@mail.ustc.edu.cn).

Digital Object Identifier 10.1109/LCOMM.2006.05008.

II. GRAPHICAL REPRESENTATION OF CYCLE GF($2^p$) CODES

Similar to graphical representation of binary cycle codes [4], a cycle GF($2^p$) code with parity check matrix $H$ of size $m \times n$ can be described by a graph $G$ of $m$ vertices and $n$ edges. Vertex $c_i$ represents the constraint node defined by the $i$th row of $H$. Edge $x_{ij}$ represents the symbol node corresponding to the $j$th column of $H$. Vertices $c_i$ and $c_k$ are connected by edge $x_{ij}$ if and only if the $j$th column of $H$ has non-zero entries at exactly the $i$th and $k$th rows.

Example 1: Assume that the parity check matrix $H$ for a cycle GF($2^2$) code is given by

$$H = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 3 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Its associated graph is shown in Fig. 1. The non-zero entries of $H$ are not presented because we are rather concerned about the code’s graphical structure. We will denote a cycle GF($2^p$) code with parity check matrix $H$ and associated graph $G$ by $C_E(G, H)$.

Assume $C$ is a cycle. We will use notation $H_C$ to denote the sub-matrix of $H$ restricted to the rows and columns indexed by the vertices and edges of $C$ respectively and call it the sub-matrix associated with $C$. Simply by permuting the rows

![](https://example.com/figure1.png)

Fig. 1. The associated graph for $H$. 

Jie Huang and Jinkang Zhu

IEEE COMMUNICATIONS LETTERS, VOL. 10, NO. 5, MAY 2006 1

1089-7798/06$20.00 c \ 2006 IEEE
and columns of $H_C$, $H_C$ can be transformed into a canonical form $H^{-}$ as shown in (1), where $\alpha_i$s and $\beta_i$s are elements from GF($2^p$).

$$H^- = \begin{bmatrix} \alpha_1 & 0 & 0 & \ldots & \beta_k \\ \beta_1 & \alpha_2 & 0 & \ldots & 0 \\ 0 & \beta_2 & \alpha_3 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & \beta_{k-1} \alpha_k \end{bmatrix} \quad (1)$$

**Definition 1 (irresolvable)** For a cycle $C$ of length $k$, if its associated sub-matrix $H_C$ is full rank, i.e., rank($H_C$) = $k$, we call the cycle irresolvable; otherwise resolvable.

A cycle is irresolvable iff the columns of $H$ indexed by the edges of the cycle are linearly independent.

### III. Linear-Complexity Encoder for Cycle GF($2^p$) Codes

Our main theorem is the following.

**Theorem 1:** Any $C_E(G, H)$ code can be encoded in linear time.

We can restrict our attention to $C_E(G, H)$ code where $G$ is a nontrivial connected graph. If $G$ has $\omega$ nontrivial connected components $G_1, \ldots, G_\omega$, and every component $G_i$, $i = 1, 2, \ldots, \omega$, can be encoded in linear time, then the code can be encoded in linear time.

**Lemma 1:** Any $C_E(G, H)$ code where $G$ is a nontrivial tree has only one codeword, i.e., the all-zero word.

*Proof:* $G$ must contain a vertex of degree one because it is a tree. Say $c$ is a vertex of degree one and $x$ is the edge that is incident with $c$. The value of $x$ must be zero to satisfy the constraint indexed by $c$. Therefore, we can delete $c$ and $x$ from the graph and the resultant graph is still a tree. Again we can affirm the existence of a vertex of degree one in the resultant graph. By induction we complete the proof. □

**Lemma 2:** For $C_E(G, H)$ code with nontrivial and connected and $H$ full row-rank, $G$ must contain an irresolvable cycle.

*Proof:* Assume the size of $H$ is $m$ by $n$. Because rank($H$) = $m$, we can find $m$ columns of $H$ that are linearly independent. The induced sub-graph of $G$ by the edges corresponding to these columns has $m$ vertices and $m$ edges. Therefore, it must contain a cycle according to Theorem II of [11] and this cycle is irresolvable. This completes the proof. □

**Lemma 3:** For $C_E(G, H)$ code where $G$ is a nontrivial connected graph and contains at least one cycle, simply by permuting the rows and columns of $H$, $H$ can be transformed into an equivalent block-diagonal upper-triangular form $H^+$ as shown in (2), where $H^-$ is as shown in (1) and $D_i$s, $1 \leq i \leq r$, are diagonal matrices.

$$H^+ = \begin{bmatrix} H^- & A_1 & 0 & \ldots & 0 \\ 0 & D_1 & A_2 & \ddots & \vdots \\ 0 & 0 & D_2 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & A_r \\ 0 & 0 & 0 & \ldots & D_r \end{bmatrix} \quad (2)$$

*Proof:* Assume $G$ contains $m$ vertices and $n$ edges and $G$ contains a length-$k_1$ cycle $C = c_1^0 \ldots c_{k_1}^0$, where $c_1^0, \ldots, c_{k_1}^0$ represent $k_1$ vertices and $x_1^0, \ldots, x_{k_1}^0$ represent $k_1$ edges. We place vertices $c_1^0, \ldots, c_{k_1}^0$ in the first tier and label edges $x_1^0, \ldots, x_{k_1}^0$ as “selected”. The vertices that connect to $c_1^0, \ldots, c_{k_1}^0$ are put in the second tier. Let $c_1^1, \ldots, c_{k_2}^1$ be the $k_2$ vertices of the second tier. For each vertex $c_i^1$, $1 \leq i \leq k_2$, we randomly pick one edge from the edges that connect $c_i^1$ to vertices of the first tier and denote it as $x_i^1$ and label it as “selected”. Except for the vertices in the $j$–th, $j \geq 3$ tier, the vertices that connect to vertices in the $j$–th tier are placed in the $j$th tier. Let $c_1^j, \ldots, c_{k_j}^j$ be the $k_j$ vertices of the $j$th tier. For each vertex $c_i^j$, $1 \leq i \leq k_j$, we randomly pick one edge from the edges that connect $c_i^j$ to vertices of the $j$–th tier and denote it as $x_i^j$ and label it as “selected”. As $G$ is a connected graph, the construction goes on until all the $m$ vertices have been included in the constructed multi-layer structure. Let $r+1$ be the number of tiers. Then $m = \Sigma_{i=1}^{r+1} k_i$. All the edges not labeled as “selected” are labeled as “unselected” and are denoted as $x_{r+2}^{j}, \ldots, x_{r+m}$. Now permute the rows of $H$ according to order $c_1^1, \ldots, c_{k_1}^1, \ldots, c_1^{r+1}, \ldots, c_{k_{r+1}}$ and permute the columns of $H$ according to order $x_1^1, \ldots, x_{k_1}^1, \ldots, x_1^{r+1}, \ldots, x_{k_{r+1}}$. The resultant matrix has the form as shown in (2). This completes the proof. □

**Example 2:** For the graph shown in Fig. 1, let $C = c_3 x_{15} c_5 x_6 c_4 x_3 c_3$, the resultant multi-layer structure is shown in Fig. 2. A possible resultant block-diagonal upper-triangular matrix is given by

$$\begin{bmatrix} 2 & 0 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 2 & 3 & 0 & 0 & 0 & 3 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 3 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 3 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 0 & 2 & 0 & 0 & 0 \\ \end{bmatrix}$$

**Lemma 4:** The solution of equation $H^+ x = b$ can be computed in linear time.

*Proof:* Assume $x = (x_1, x_2, \ldots, x_k)^T$ and $b = (b_1, b_2, \ldots, b_k)^T$. The solution of equation $H^+ x = b$ can be
computed as follows.

1. \( z_1 = b_1; z_i = s_{i-1}z_{i-1} + b_i, \ i = 2, 3, \ldots, k; \)
2. \( y_k = (1 + s_1s_2 \ldots s_k)^{-1} \cdot z_k; \)
   \( y_i = z_i - s_1s_2 \ldots s_{i-1}y_{k}, \ i = 1, 2, \ldots, k - 1; \)
3. \( x_i = \alpha_i^{-1}y_i, \ i = 1, 2, \ldots, k; \)
   where \( s_i = \alpha_i^{-1}\beta_i, \ i = 1, 2, \ldots, k. \)

Assume the coefficients have been stored before computing. Then the computation complexity is \( 2(k - 1) \) additions, \( 2(k - 1) \) multiplications, and \( 2k \) divisions over \( GF(2^p) \). This completes the proof.

Now we can give a proof of our main theorem.

**Proof of theorem 1:** As shown, we just need to verify the fact for the case where \( G \) is a nontrivial connected graph. If \( G \) is a tree, Lemma 1 shows that the encoding can be accomplished in linear time. If \( G \) is not a tree, \( G \) must contain a cycle. Lemma 3 shows that \( H \) can be transformed into a form \( H^+ \) as shown in (2) simply by permuting the rows and columns of \( H \). Denote the weight of the row of \( H^+ \) indexed by \( c^j_i \) as \( w_i^j \). Then \( 2n = \sum_{j=1}^{r+1} \sum_{i=1}^{r+1} w_i^j \). The rank of \( H \) can be \( m - 1 \) or \( m \).

**Case 1** \( \text{rank}(H) = m - 1. \)

The dimension of the code space is \( n - m + 1 \) and \( C \) must be resolvable. There must be a linearly dependent row in the first \( k_1 \) rows of \( H^+ \). We can remove this redundant row without changing the underlying code structure. Say, we remove the first row of \( H^+ \). Move the \( k_1 \)th column of \( H^+ \) to the rightmost and denote the resultant matrix as \( H^u \). \( H^u \) is an upper-triangular matrix. Let symbols corresponding to the last \( n - m + 1 \) columns of \( H^u \) be information symbols and others be parity symbols. Encoding with \( H^+ \) can be accomplished by a backward recursion. When a row of weight \( f \) is used to determine the value of a parity symbol, at most \( f - 1 \) additions, \( f - 1 \) multiplications, and one division over \( GF(2^p) \) are needed. Therefore, the encoding complexity is at most \( \sum_{j=1}^{r+1} \sum_{i=1}^{r+1} (w_i^j - 2) = 2n - 2m \) additions, \( \sum_{j=1}^{r+1} \sum_{i=1}^{r+1} (w_i^j - 1) = 2n - m \) multiplications, and \( \sum_{j=1}^{r+1} \sum_{i=1}^{r+1} 1 = m \) divisions over \( GF(2^p) \). In this case the encoding can be accomplished in linear time.

**Case 2** \( \text{rank}(H) = m. \)

The dimension of the code space is \( n - m \). Lemma 2 guarantees the existence of at least one irresolvable cycle. Let \( C \) be an irresolvable cycle. Then the first \( m \) columns of \( H^+ \) are linearly independent. Let the \( n - m \) symbols \( x_1^{r+2}, \ldots, x_{n-m}^{r+2} \) be information symbols and others be parity symbols. Encoding with \( H^+ \) can be accomplished by a backward recursion. To determine the \( m - k_1 \) parity symbols \( x_1^r, \ldots, x_{k_1}^r, \ldots, x_1^{r+1}, \ldots, x_{k_1}^{r+1} \) at most \( \sum_{j=1}^{r+1} \sum_{i=1}^{r+1} (w_i^j - 2) \) additions, \( \sum_{j=1}^{r+1} \sum_{i=1}^{r+1} (w_i^j - 1) \) multiplications, and \( \sum_{j=1}^{r+1} \sum_{i=1}^{r+1} 1 \) divisions over \( GF(2^p) \) are needed. The last step for computing \( x_1^r, \ldots, x_1^{r+1} \) requests solving an equation like \( H^{-1} \cdot x = b \). Lemma 4 shows that at most \( \sum_{i=1}^{r+1} (w_i^1 - 2) + k_1 - 2 \) additions, \( \sum_{i=1}^{r+1} (w_i^1 - 1) + k_1 - 2 \) multiplications, and \( 2k_1 \) divisions over \( GF(2^p) \) are needed. Therefore, the encoding complexity is at most \( 2n - 2m + k_1 - 2 \) additions, \( 2n - m + k_1 - 2 \) multiplications, and \( m + k_1 \) divisions over \( GF(2^p) \). In this case the encoding can be accomplished in linear time. This completes the proof.

**IV. Conclusion**

In this letter, we proposed a linear-complexity encoding algorithm for any cycle \( GF(2^p) \) code through graph analysis. Compared with the method given in [9] for binary cycle codes, our method works for both binary and nonbinary cycle codes. This is quite desirable in many applications.

**References**


